DOC. TYPE 2
MGT. CODE SE

JSC-10019
PART I
7 APRIL 1976

NASA CR-147864

Screwworm Eradication Data System (SEDS) Preprocessor Program Documentation

(NASA-CR-147864) SCREWWORM ERADICATION DATA SYSTEM (SEDS) PREPROCESSOR PROGRAM DOCUMENTATION, PART 1 (Aeronutronic Ford Corp.) 129 p HC A07/MF A01 CSCL 06C N77-23713
Unclas
G3/51 26699

Contract NAS 9-1261

DRL LI 2.20

prepared for

National Aeronautics and Space Administration

Lyndon B. Johnson Space Center

Aeronutronic Ford
Aeronutronic Ford Corporation
Space Information Systems Operation
1002 Gemini Avenue
Houston, Texas 77058

Doc. Type 2
Mgt. Code SE

JSC-10019
Part I
7 April 1976

SCREWWORM ERADICATION DATA SYSTEM (SEDS) PREPROCESSOR PROGRAM DOCUMENTATION

Contract NAS 9-1261

DRL LI No. 2.20

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas

H. M. Campbell

H. M. Campbell, Supervisor
Support Systems Section

HG Johnson

H. G. Johnson, Manager
Software Systems Department

CW Abbitt

C. W. Abbitt, Manager
System Engineering Activity

AERONUTRONIC FORD CORPORATION
AERO-COMM ENGINEERING SERVICES DIVISION
SPACE INFORMATION SYSTEMS OPERATION
1002 GEMINI AVENUE
HOUSTON, TEXAS

FOREWORD

This document is provided by the Space Information Systems Operation (SISO) in accordance with the requirements of Task Order (TO) P-6Q00 as established under modification No. 201 of Contract NAS 9-1261, Schedule V, and DRL Line Item 2.20.

Parts II, III and IV of this document will be forthcoming under separate covers at a later date. They will deal with, respectively, SEDS applications software, the program operator's guide, and the SEDS program listings.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

LIST OF FIGURES

LIST OF TABLES

SECTION 1

INTRODUCTION

1.1 PURPOSE

To achieve the capability required by the Screwworm Eradication Data System (SEDS) and to process the 14-track analog tapes supplied by the National Oceanic and Atmospheric Administration (NOAA), the existing Earth Resources Preprocessor Software Subsystem (ERPSS) was enhanced. This document gives a brief overview of the ERPSS and furnishes detailed descriptions of only those computer program components that were modified to meet the SEDS requirements.

1.2 GENERAL DESCRIPTION

The function of the SEDS preprocessor is to format data digitized from NOAA-supplied 14-track analog tapes for output to a 9-track computer-compatible tape (CCT) in universal format and/or the 525-line interactive color display. The system is initialized via operator inputs which define data format and output medium. The passes made by the SEDS preprocessor to achieve the desired final results are described below.

A. Screening Pass. The first pass outputs the raw data to the 525-line interactive color display. The operator controls the system from the display console and has the option of defining an area and tagging points within this area on the display. When the operator is satisfied with the viewed image and area defined, the run is terminated and the results evaluated for initialization of the next pass.

B. Edit Pass. The second pass follows the screening of the data and formats and outputs data to a 9-track tape for that area defined in the first pass. The satisfactory completion of this pass terminates the preprocessor portion of a total SEDS production run.

C. Edit While Screening Pass. This pass allows the operator a capability to screen and edit simultaneously if he has the necessary information for initialization of the preprocessor in this dual mode.

1.3 REFERENCE DOCUMENTS

SISO-TR545, *Earth Resources Production Processing System Preprocessor Subsystem Software Design Document*, dated 28 March 1974, and the applicable documents referenced within it, provide a total description of the standard ERPSS.

SECTION 2

SYSTEM SOFTWARE

2.1 GENERAL

The system software for the SEDS preprocessor consists of two sections, the PDP11 Disk Operating System (DOS) and its associated system programs, and the system software provided by SISO for the preprocessing task.

2.2 DEC SOFTWARE

The software supplied by Digital Equipment Corporation (DEC) consists of the DOS Batch Monitor and the various supporting system programs listed below.

- MACRO-11 Assembler
- FORTRAN IV Compiler
- FORTRAN Library and Math Routines
- EDIT-11 Text Editor
- ODT-11R Online Debugging Program
- PIP-11 File Utility Package
- LINK-11 Linker
- LIBR-11 Librarian
- ROLLIN System Generation Program
- VERIFY Disk Structure Verification Program
- FILCOM File Compare Program
- FILDMP File Dump Program.

2.2.1 DOS Batch Monitor. The DOS Monitor provides convenient user access to system programs and a means to accomplish input/output (I/O) on three different levels, ranging from direct access to device drivers to a fully formatted capability. It also provides two-way communication with the user/operator as described below.

A. Console/Software Control. The user communicates with the Monitor through programmed instructions or requests within his program, and commands via the console keyboard. Programmed requests allow the user to utilize the DOS I/O transfer facilities, specifying the location of the data, its destination, and its format. Other requests enable the user to access system parameters, to monitor, and to control to some degree the system status.

B. Batch Processing. The PDP11 Batch Monitor is another method for controlling computer operations. Commands located on cards and fed through the card reader dictate operations to be performed. Command cards are arranged in a deck along with source programs and/or data. By placing several decks together, a "job stream" is formed. The system reads each command card in the stream, performs the requested operation, and then automatically processes each successive operation until the job stream is complete. This allows programs to be prepared offline, leaving computer resources for more critical tasks.

2.2.2 MACRO-11 Assembler. This program operates under the DOS Monitor and allows the user to write sources of symbolic programs in numbers and letters which are meaningful to him. Source programs are generated by using the EDIT-11 Text Editor Program or generated offline, and are assembled by the MACRO-11 Assembler into object modules which are processed by the LINK-11 Linker. This in turn produces a load module which is loaded by the Monitor for execution. MACRO-11 also produces a complete octal/symbolic listing of the program and a cross-reference symbol table. It has the capability to generate recurring coding sequences with a single statement through the use of macros. Other characteristics of MACRO-11 include the following.

- Selective assembly pass functions
- Device and file name specification for pass functions
- Error listing on command output device
- Alphabetized, formatted cross-reference table
- Relocatable object modules
- Global symbols for linking between object modules
- Conditional assembly directives
- Program sectioning directives.

2.2.3 FORTRAN IV Compiler. This program operates under the DOS Monitor. FORTRAN IV language is problem-oriented and allows the user to express computations in an algebraic form. A FORTRAN program comprises statements in an easily readable form. Computational elements are expressed in a notation similar to that of standard mathematics, and commands are descriptive of the functions they perform. The source program is prepared by using either EDIT-11 or another text editor, EDIT-73, or is prepared offline. The FORTRAN IV Compiler compiles the source program and produces an object code which is processed by the LINK-11 Linker to produce a load module. The load module may then be loaded by the DOS Monitor upon request. The compiler does not generate assembly code to be subsequently assembled by the user with the MACRO-11 Assembler. Object code is produced directly from FORTRAN source code.

2.2.4 FORTRAN Library and Math Routines. The FORTRAN Library as used with the PDP11 FORTRAN IV Compiler consists of four subroutines which enable the user to:

- Change device assignment
- Acquire the current date

- Set the maximum allowable occurrence counts of specified classes of errors

- Dump specified areas of core in a specified format.

Standard FORTRAN IV function subprograms are also included in the FORTRAN Library. These subprograms provide all of the mathematical and trigonometric functions that will be required by the FORTRAN IV Compiler.

2.2.5 EDIT-11 Text Editor. EDIT-11 is a system program which operates in the DOS environment. Operated by user commands from the console keyboard, it will read ASCII files from any device, make directed changes, and write the results on any device. In addition to basic editing functions, EDIT-11 provides for command macros and multiple input and output files. The command macro capability allows the user to make multiple changes to repetitious ASCII character sequences with only one EDIT-11 command. EDIT-11 makes editing simple and flexible due to the large number of commands available. Each command consists of one or more letters, and commands may be strung together to form a command string with appropriate arguments and test objects included. The command string is then executed by EDIT-11, acting on the contents of an internal buffer in which the input source is stored. The editing process can be divided into three sequential actions:

- Reading input text into internal buffer

- Changing the text stored in the buffer

- Output of the revised text to designated device.

2.2.6 ODT-11R Debugging Program. ODT-11R is a system program operating under the DOS and is used to aid in debugging linked object programs. The user links ODT-11R with other modules in his program via the LINK-11 Linker and interacts with these through his keyboard to:

- Print the contents of any location for examination or alteration

- Run any or all portions of the object program utilizing the breakpoint feature
- Search the object program for specific bit patterns
- Search the object program for words which reference a specific word
- Calculate offsets for relative addresses
- Fill a block of words or bytes with a designated value.

The user may also make minor corrections to the program online during the debugging session. The program may then be run under the control of ODT-11R to verify any changes that were made.

2.2.7 PIP-11 File Utility Package. This is a system program which performs file operations for the PDP11 DOS. Some of these operations are file transfers between devices, directory listings of disk or magnetic tape, deleting files, and renaming files. The major functions performed by PIP-11 are:

- Allocating a contiguous file on disk
- Listing the directory of a disk or magnetic tape, printing only file names.
- Listing the directory of a disk or magnetic tape, printing file names, file size, creation date, and protection code
- Deleting a file or group of files
- Changing the protection code of a file
- Renaming or changing the name of a file
- Initializing or deleting a magnetic tape directory
- Transferring a file or group of files from one device to another
- Merging a group of input files into one output file.

2.2.8 LINK-11 Linker. LINK-11 is a system program operating in the DOS environment, used to link and relocate user programs assembled by MACRO-11 or compiled by the FORTRAN IV Compiler. LINK-11 enables the user to assemble each module of his program separately without having to assign an absolute address for each segment at assembly time. The object module from each assembly of compilation is processed by LINK-11 to:

- Create a library
- Update a library
- Insert, replace or delete one or more modules in a library
- List the contents of the library
- Delete an entire library.

2.2.9 ROLLIN System Generation Program. ROLLIN is a utility program designed for the simple and rapid installation of the DOS Monitor on configurations having magnetic tapes. It is a stand-alone program which is used to facilitate the dumping and restoration of data between disk storage and magnetic tape devices.

2.2.10 EDIT-73 Line-Oriented Text Editor. EDIT-73 is a line-oriented text editing program which allows a user to update source programs on the PDP11. Written in FORTRAN, EDIT-73 allows the user to list, insert, delete, or replace lines of any source program.

2.2.11 VERIFY Disk Structure Verification Program. VERIFY's basic application is to rapidly verify whether or not the file structure on a given device is good. If problems are encountered, VERIFY provides diagnostic information which can be interpreted to determine the best corrective action. The causes of such problems range from hardware malfunctions to system program errors and user errors. VERIFY will not correct these errors, but will aid the user in isolating the causes.

2.2.12 FILCOM File Compare Program. FILCOM provides the user with a means for comparing two ASCII files. The user then has the option of creating a file listing those differences located during the comparison run. This program facilitates location of modifications in source modules and identification of the most current program.

2.2.13 FILDMP File Dump Program. FILDMP is a system utility program designed to dump all of or specific blocks of a file for visual inspection. The dump can be directed to any device capable of handling ASCII output.

2.3 PRODUCTION PROCESSING SYSTEM SERVICE PROGRAMS

The Production Processing System service routines described in the following paragraphs will be invoked through emulator trap (EMT) instructions in an analogous fashion to those service routines which are included in the DOS Monitor. The EMT handlers provided by DEC follow certain operating conventions and are physically organized on the system-disk in a special way. For this reason, any new EMT handlers added must be consistant in both operating requirements and system organization with those DOS service routines already provided. A list of the conventions to be followed in adding the new handlers is given below (these requirements were assimilated from DEC 11-OMONA-A-D, *PDP-11 Disk Operating System Monitor (Version V008A) System Programmer's Manual*.

A. The new service routine must restore user registers before exiting calling program.

B. The following values will be passed to the new service routine in the general registers by the DOS EMT handler:

- R1 = Stack address of last call parameter pushed by the user program
- R2 = Address of the EMT call in user program
- R5 = Address of the processor status register
- RO = 0.

C. Coding of the service routine must be position-independent.

D. Registers or processor stacks may be freely used internally by a module or between modules as long as they are restored to the original program state when control is returned to that program.

E. If the code is not re-entrant, it must be protected.

F. Each new service must be allocated an EMT call code.

G. The new services will not be "swappable," but will be automatically called into core whenever a production job is run. They do not have to follow the conventions established for potentially "swappable" Monitor service routines.

H. The Monitor Resident Table (MRT) source in the DOS must be modified with the global names of the new service routines.

I. The MRT reference list of EMT entries must be expanded in the initialization routine of DOS.

J. New service routines must be added to the Monitor's library.

The following paragraphs give a brief description of the Production Processing System service programs that are utilized by the preprocessor while processing in the SEDS environment.

2.3.1 <u>14-Track Tape Handler (FTH)</u>. FTH provides the software interface between the 14-track tape hardware and the user of the 14-track tape. The handler initiates all commands to drive the 14-track tape (as specified by the user program), receives all interrupts initiated by the device, and returns device status to the user program as requested. The following two points are the only enhancements that were required for SEDS implementation.

A. In the initialization service portion of FTH, the first word address (FWA) for each buffer area is converted to an 18-bit virtual address. Under normal operations, each scientific data output (SDO) is associated with two buffers having the same physical address. But when these FWA's are converted, parameter address registers (PAR's) are modified to map the buffers into different virtual areas. This process has been enhanced to branch around the modification of the PAR's during the buffer address conversions for a SEDS screening run. For this pass only, both buffers must have the same virtual address to enable use of the display's monitor mode. See figure 2-1 for the flow of this FTH software modification.

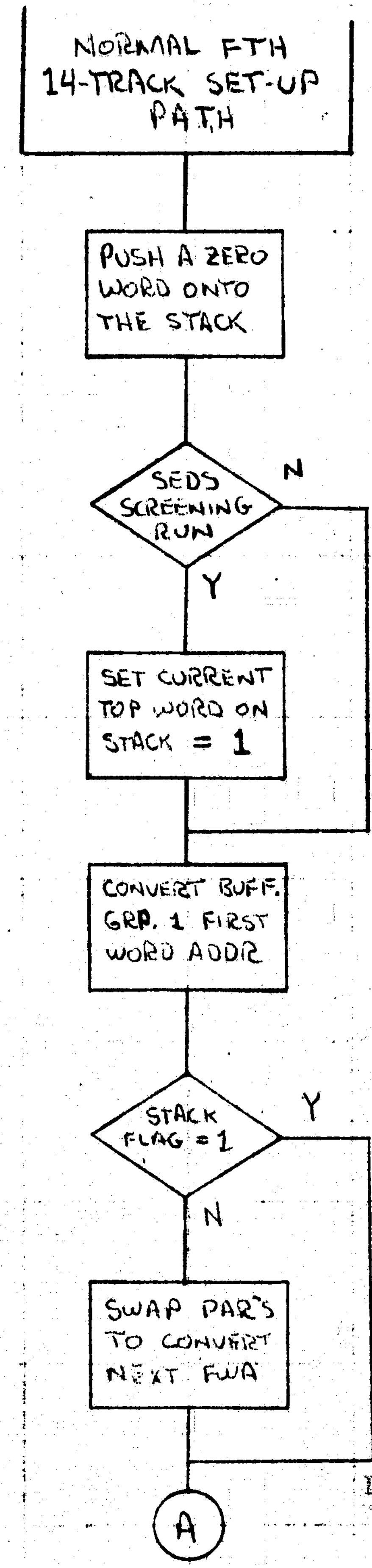
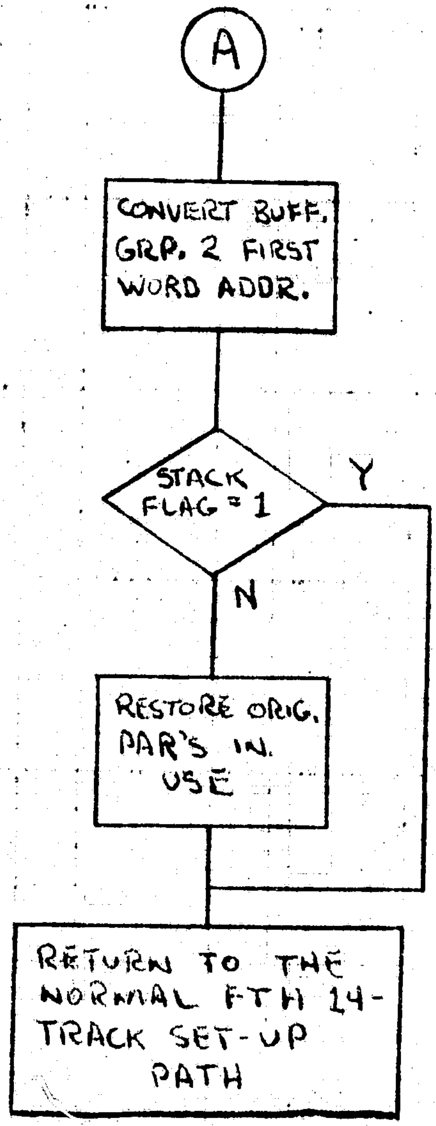


Figure 2-1 FTH Software Modification Flow



ORIGINAL PAGE IS OF POOR QUALITY

Figure 2-1 (Cont'd)

B. For SEDS processing, the data is patched so that it always plays through the interface hardware for track 12. The tape interface register addresses to control track 12 are used by the SEDS logic, but some of the parameters expected are not the same as for a normal run. These values ultimately are only transferred to the hardware registers from the T14HAN Table by FTH. The modified values for SEDS are put in the table by the Initialization Routine, TINIT. Figure 2-2 gives the register addressed for track 12 and the values expected by the SEDS logic.

2.3.2 Display Handler (VT05). VT05 interfaces the operator interface programs on the preprocessing system with the VT05 alphanumeric keyboard and display device hardware. VT05 receives requests to output character strings to the display, enables interrupts for receiving keyboard input, and notifies the proper programs to process data input.

2.3.3 Program Scheduling Service (SCH). SCH allows one core-resident program to schedule another core-resident program for execution at a predefined priority level. In addition, the scheduler will allow one parameter to be passed to the scheduled program. SCH, PIRP, and EXH are linked to one another in the scheduling of core-resident programs. Routines entered via SCH should be exited only via EXH.

2.3.4 Program Interrupt Request Processor (PIRP). This interrupt processing routine is activated by SCH. PIRP receives control by the setting of a program interrupt request (PIR) bit in the register located at virtual address 177772_8. It transfers control to the higher priority job if one is scheduled, or to the next highest when the previous job exits through EXH.

2.3.5 Program Exit Handler Service (EXH). EXH is called to process program exit requests for programs scheduled by SCH. EXH will dequeue and dispatch any queued program request for the current active processor priority level. SCH, PIRP, and EXH are all linked to one another in the scheduling of core-resident programs. Routines entered via SCH should be exited only through EXH.

764500*	STATUS REGISTER 1
764502*	COMMAND REGISTER
764504*	FWA BUFFER GROUP 1
764506	NOT USED
764510*	FWA BUFFER GROUP 3
764512	NOT USED
764514[†]	START SAMPLE/PIXEL COUNT
764516[††]	STOP SAMPLE/PIXEL COUNT
764520	NOT USED
764522	NOT USED
764524[†††]	WORDS/FRAME
764526*	COMMAND REGISTER
764530*	STATUS REGISTER 2

*UNCHANGED FOR SEDS

[†]764514 (START SAMPLE/PIXEL COUNT) = 12-BIT BINARY VALUE THAT SPECIFIES SAMPLE COUNT WHICH ENABLES DATA TRANSFER TO DATA BUFFER

[††]764516 (STOP SAMPLE/PIXEL COUNT) = 12-BIT BINARY VALUE THAT SPECIFIES SAMPLE COUNT WHICH DISABLES DATA TRANSFER

[†††]764524 (WORDS/FRAME) = 16-BIT BINARY VALUE THAT SPECIFIES NO. OF WORDS IN SERIAL INPUT BIT STREAM BETWEEN SYNC CODES; NO. OF WORDS/FRAME IS ONE GREATER THAN BINARY VALUE ENTERED, FOUND BY THE FOLLOWING:

$$\text{WRDS/FRM} = [(B-A)+13]\ 2-1$$

WHERE:

A = START SAMPLE COUNT
B = STOP SAMPLE COUNT

Figure 2-2 SEDS 14-Track Interface Registers

2.3.6 Programmable Timer Service (PTS). PTS allows a core-resident program to schedule another core-resident program after a specified number of 1-second clock intervals have elapsed. When the specified delay reaches zero, the program request is passed to SCH for further processing. PTS keeps an accumulative total of the clock interrupts used in internal time.

2.3.7 Nonstandard Bucode Magnetic Tape Driver for Write Only (DIRTY$). DIRTY$ provides a fast interrupt-driven, write-only capability for magnetic tape. This is required to keep pace with the high transfer rates of the 14-track tape hardware. There are three operations possible with this routine: device initialization, write record, and write end-of-file (EOF). DIRTY$ is divided into two parts, the interrupt service module (DIRT$M) and the noninterrupt user service portion (DIRT$Q). DIRT$M contains the only modifications required to this program for SEDS implementations.

The number of records per data set written must be monitored as the write-complete interrupts are processed by DIRT$M. This is done so that the last two bytes of the second record can be overlaid with the record number of the third record before it is logged. This is required because the two data types, infrared and visible, are placed in a single buffer in a contiguous manner and they have to be separated and logged as individual records within a data set. Figure 2-3 gives a graphic representation of this situation. See figure 2-4 for the flow of DIRTY$ enhancements for SEDS.

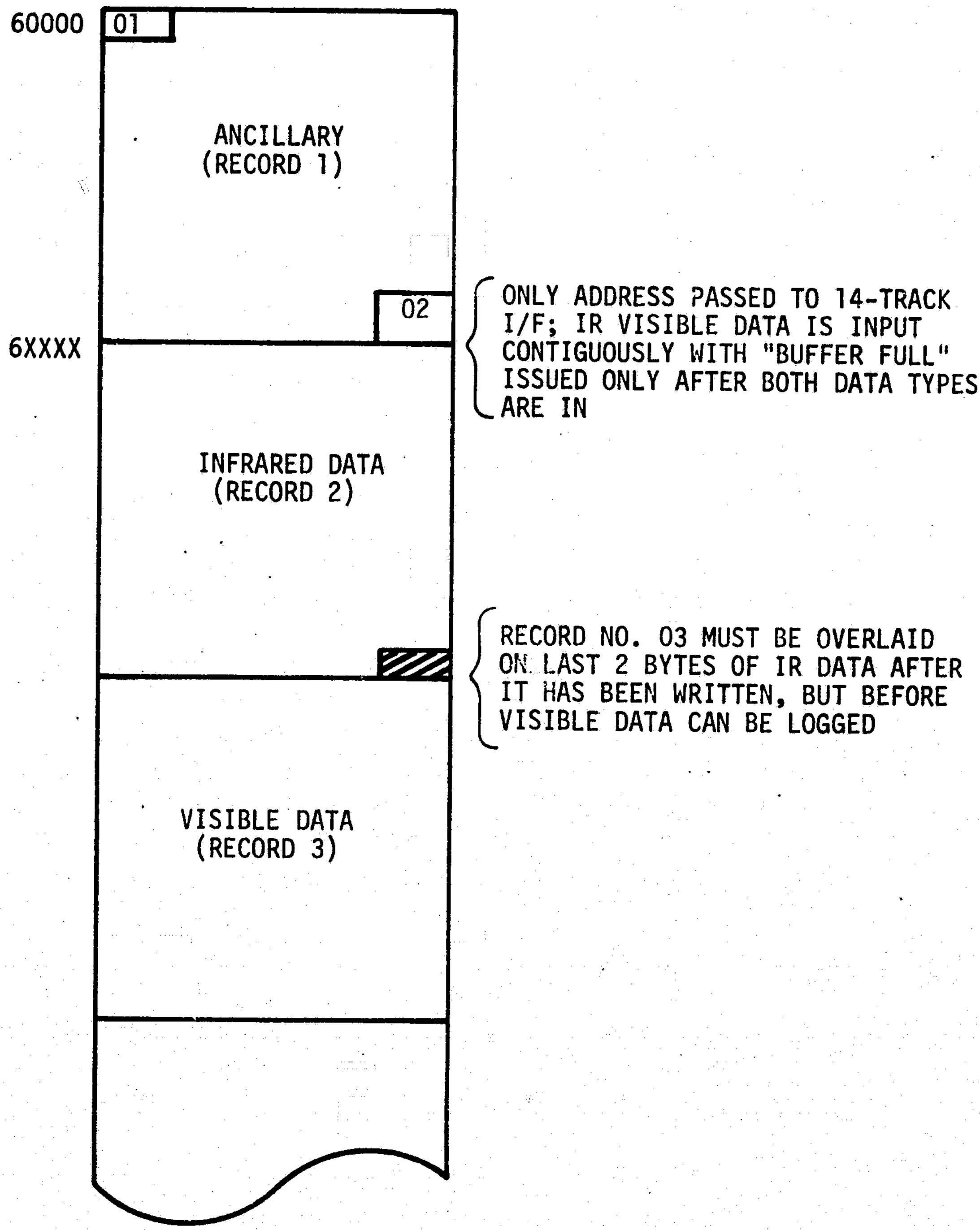


Figure 2-3 Screwworm 14-Track Data Buffer Structure

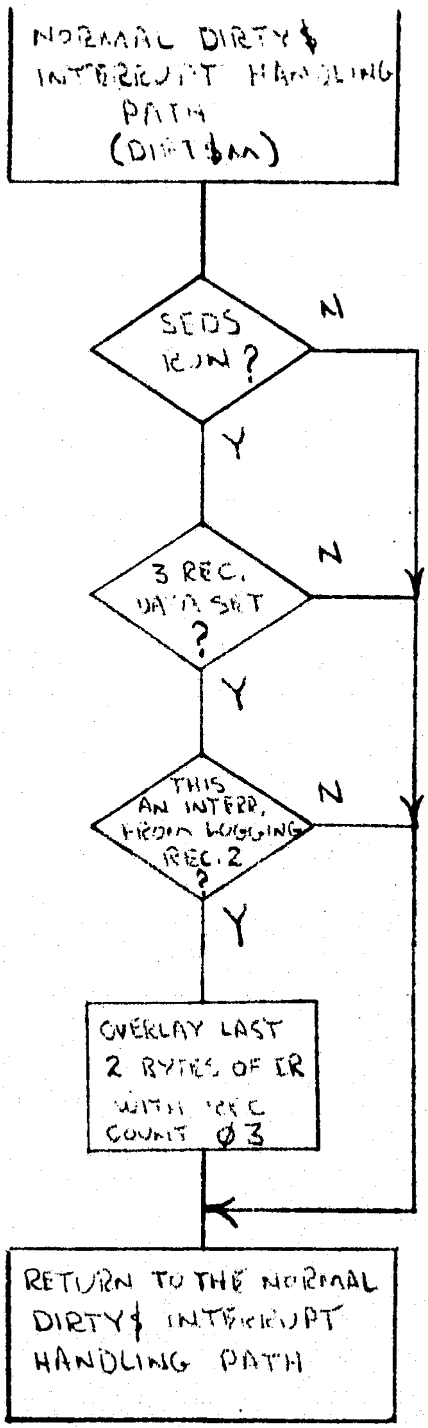


Figure 2-4 Flow of DIRTY$ Enhancement for SEDS

2.4 PREPROCESSOR MEMORY SEGMENTATION SERVICES

The following paragraphs discuss the method used to implement memory segmentation in the preprocessor. Figure 2-5 illustrates the physical core layout and attempts to show how the page address registers are used. This memory segmentation scheme has not been modified from its current configuration by any of the modifications installed to satisfy the SEDS functional requirements.

A. Criteria for Utilization of Memory Segmentation Hardware in the Preprocessor. In selecting a method for effectively utilizing the PDP11/45 memory segmentation hardware, it became apparent that the method chosen should meet certain minimum criteria, as follows:

- Minimum implementation time
- Maximum possible program and data areas
- Minimum restrictions on all preprocessor programs
- Minimum paging; i.e., as few as possible calls to memory segmentation service programs by preprocessor programs needing access to areas of physical core not active
- Fewest possible modifications to DOS so that DOS services would be available to preprocessor programs
- Minimum number of programs to service the memory segmentation hardware
- Easiest possible implementation, requiring minimal software/software interfaces.

B. Selected Method. The following paragraphs describe the advantages of the method selected for managing memory in the preprocessor. This method meets all of the requirements listed above.

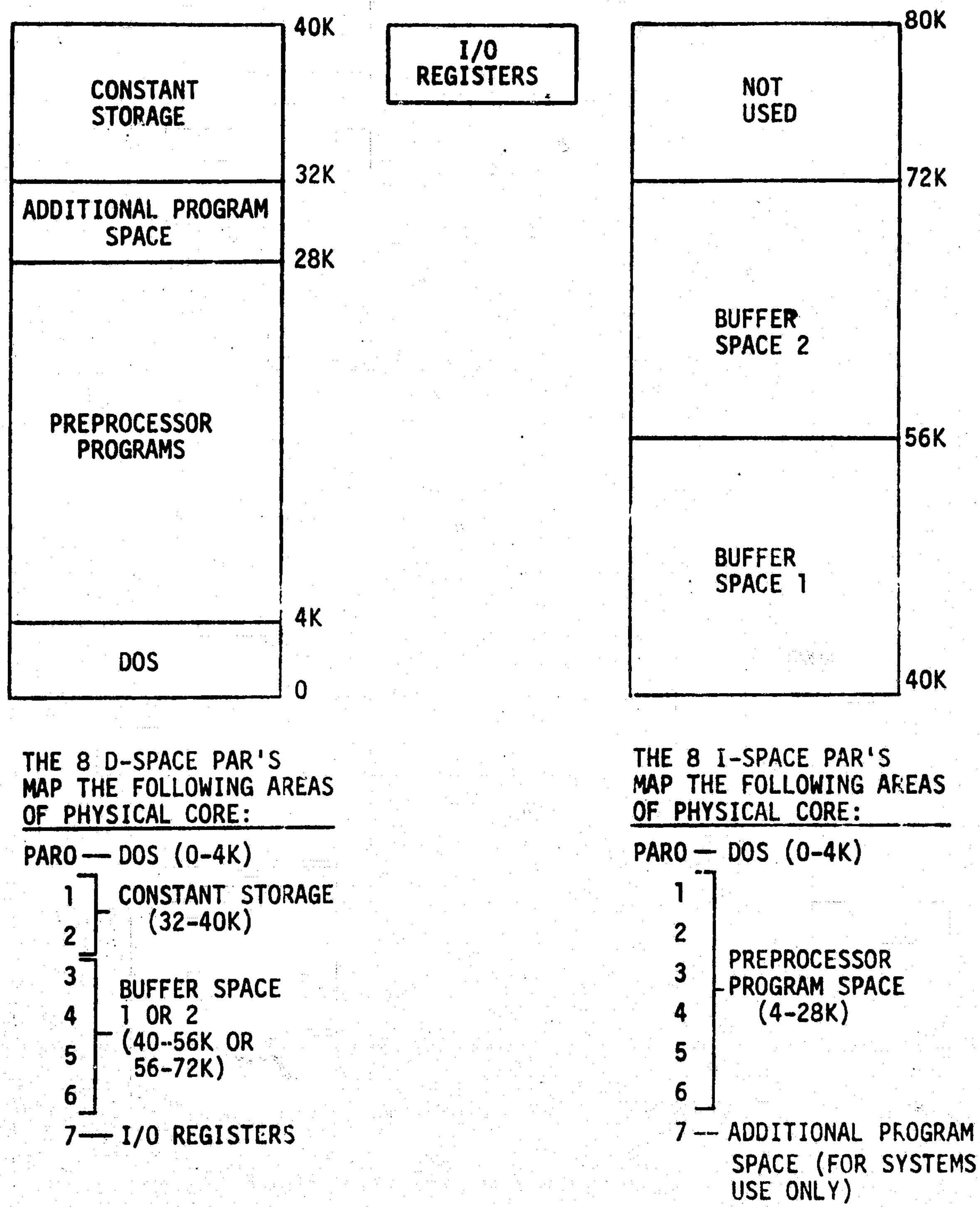


Figure 2-5 Physical Core Layout in the Preprocessor Primary Computer

1. This method provides the maximum program and data area possible, employing memory segmentation I (instruction) and D (data) space which allows virtual addressing of 64K words of memory. Of the 64K words available, only 60K words are actually used in order to accommodate the DOS. Figure 2-5 illustrates the physical memory layout using this scheme. Of the 60K words available to the preprocessor, 24K words will be available to data and buffers, and 28K words will be available to preprocessor programs. It is highly desirable, however, that preprocessor programs total less than 24K words to facilitate loading, since the present DOS loader cannot load programs above 28K. Of the 24K words available for data and buffers, 16K words are allotted for buffers and 8K words contain preprocessor constant storage. Programs occupy I-space; data and buffers occupy D-space. One 4K word block of D-space is occupied by device I/O registers. One 4K word block of I-space coincides with a 4K word block of D-space and contains the DOS. Systems programs will occupy a physical block of 28-32K words.

2. The preprocessor programs are restricted in only one way -- instructions and data must be kept separate (flags, constants, etc.). This means that all preprocessor programs must share a common area designated for constant storage. This is most easily accomplished by creating a single program containing all constant storage for all preprocessor programs; with 8K words available for this, constant storage block size limitations should not be a factor.

3. The memory segmentation registers may be initialized once and changed thereafter only on buffer-full conditions when processing synchronous data in the preprocessor. Furthermore, the buffer-full condition necessitates only the changing of four memory segmentation register values. Only one preprocessor program (ALARM)

is required to interface directly with memory segmentation service programs in addition to the preprocessor initialization programs. An additional program (EMTFIX) is required to provide segmentation services for DOS routines.

C. Required DOS Modifications. It is estimated that the DOS may be used with only minimal modification, which is required to accommodate I-D space overlapping. Prior to enabling the memory segmentation hardware, the DOS will be allowed to operate in a 28K environment to load preprocessor programs. DOS I/O should be used with caution in the preprocessor because DOS I/O routines currently do not use the 18-bit buffer addressing facility, and I/O with 16-bit addresses may overlay preprocessor programs. DOS modification or extensions are required to resolve the following problems.

1. Preprocessor programs must be loaded so that the low load address is 20000_8. Data and systems programs must be loaded separately and moved during the initialization process.

2. Preprocessor routines requiring DOS I/O services which do not utilize an intermediate monitor buffer (e.g., TRAN requests) will necessitate modifications to DOS I/O routines. This may be foregone by using only read/write level requests.

3. The EMT handler must use segmentation services to fetch the EMT code from I-space.

D. Required Interfaces. Minimal software/software interfaces are required. Only ALARM, in addition to the initialization programs, will call PSI or PSS, and calls will require no parameters. (PSS will be called by ALARM to process synchronous data from the 14-track tape; asynchronous data will require no paging.) CVP will be called by FTH, HDH, and HIDENT with only one parameter -- the 16-bit address to be converted.

2.4.1 Preprocessor Segmentation Initialization (PSI). PSI provides two services to the preprocessor, as described below.

A. It enables the memory segmentation hardware and initializes the memory segmentation hardware during preprocessor initialization (this provides 60K words of additional addressable memory for utilization by the preprocessor software).

B. It disables the memory segmentation hardware during preprocessor termination, thereby putting the system in a ready state under the DOS.

2.4.2 Preprocessor Segmentation Service (PSS). PSS provides a buffer paging facility for preprocessor data formatting programs processing synchronous data, the category that SEDS data processing falls into. Under the preprocessor memory segmentation scheme, two buffers (each 16K words long) may be physically available, but only one buffer group may be addressed. PSS provides a means for altering the contents of the appropriate memory segmentation PAR's, allowing a program to address the second buffer group.

2.4.3 Convert Virtual to Physical Routine (CVP). CVP is provided as a memory segmentation service routine to convert 16-bit virtual addresses to 18-bit physical addresses, using the memory segmentation registers (PAR's) to simulate the conversion performed by the memory segmentation hardware. CVP is useful to routines such as I/O handlers requiring an 18-bit buffer address (e.g., HDH).

2.4.4 DOS EMT Handler Modifications (EMTFIX). EMTFIX is provided to allow the DOS-resident EMT handler (dispatch routine) and the DOS EMT to call code from I-space, which requires a temporary remapping. While implementing the SEDS enhancements, minor DOS problems created as a result of using memory segmentation with I- and D-space may also be corrected in EMTFIX.

2.5 PREPROCESSOR INITIALIZATION

The preprocessor initialization is accomplished by an operating procedure and a set of initialization programs. The procedure and routines that are described in the following paragraphs are those that affect the initialization of the preprocessor for processing of the SEDS data.

A. The operator calls the BATCH Monitor into the computer via the switch register booting procedure. At the completion of the batch load, a message will be printed on the DEC writer notifying the computer operator that BATCH is now ready to accept an operator command. At this point, the operator will enter via the operator keyboard the following commands:

```
$RUN
$RUN MVCORD
$RUN AMAGE
```

B. The processing software takes control and reads a job card deck, then displays the initialization parameters on the operator's VT05. The operator may then review the proposed processing configuration and accept or override any of the parameters via a VT05 input. When directed to proceed, the actual 14-track processing begins, leaving the operator with only the option of halting the run.

2.5.1 <u>System Software Move (SYSD)</u>. SYSD moves the SEDS Preprocessor System service and memory segmentation routines, loaded in low core, to upper core, beginning at 160000_8 of I-space. These modules are originally linked at 140000_8 before being moved; this makes it important that they utilize completely position-independent code (PIC)

2.5.2 <u>Move Data Space Modules (MVCORD)</u>. MVCORD envokes memory segmentation and moves the preprocessor data space object modules, originally loaded at 20000_8 in lower core, to upper core beginning at 200000_8. Due to the segmentation and I/D-space scheme, all data constants, buffers, tables etc.; must be kept separate from executable code and linked in this module.

2.5.3 Execute Processing Software (AMAGE). This module begins the actual processing of the 14-track data. Each routine in AMAGE can be placed in one of the following processing categories.

A. Initialization. This area includes the remaining system initialization such as enabling memory segmentation and setting EMT/interrupt vectors. Also, the initialization parameters for processing are input via card and/or VT05 and acted upon to complete all required setup.

B. Processing. By the completion of the initialization phase, the 14-track has already begun inputting data. The processing software enters an idle loop following initialization until the first buffer-full interrupt is received from the 14-track hardware. Each time an interrupt is received, basic data quality checks are made, the data is formatted for output, and output is initiated. The processing software then returns to the idle loop until another interrupt is issued. The first buffer-full interrupt triggers the data quality display and logging software, which reschedules itself for execution at a fixed interval until a run termination condition is created. Upon completion of 14-track processing, the system outputs the data quality and summary reports, frees any disk area allocated during the run, and returns to the DOS Monitor.

SECTION 3

PROCESSING SOFTWARE

3.1 GENERAL DESCRIPTION

The principle of this software is to process data emanating from a 1-inch wide, 14-track magnetic tape. For SEDS, this is a NOAA-supplied analog tape. The processing consists of reformatting the data, checking the quality of the data, and for SEDS, outputting the data to the 525-line interactive color display (screening run) and/or a 9-track CCT in universal format (edit run). These SEDS runs are accomplished within the basic operational environment of the Preprocessing Software Subsystem of the Production Processing System.

The functions performed by the processing software are implemented in separate computer program components, each individually executed by the Monitor. There are five sections, as outlined below, which comprise the processing software.

A. Initialization. This section does all common initialization for all data format processing, as follows.

- Completes any required system setup
- Inputs and validates processing initialization parameters
- Initializes common tables per setup parameters
- Initializes 14-track interface and begins data transfer.

B. Imagery Processing. This section is self-contained and performs the following tasks.

- Validates input data
- Updates common tables for other section of the processing software

- Performs any data formatting required for correct output to magnetic tape.

C. Nonimagery Processing. This section is also self-contained, and performs the same tasks as the imagery processing software, but for the nonimagery data.

D. Processing Support Software. This section provides the following services to support both the imagery and nonimagery processing sections.

- Software error checking
- Interface with output tape driver
- Time service
- Error tables builder.

E. Processing Utility Programs. These are standalone programs to examine and/or dump the data output by the processing software.

The above component sections provide the main data processing for the Preprocessing Subsystem. The imagery processing and nonimagery processing sections are mutually exclusive of one another. None of the nonimagery routines are utilized in the SEDS environment, so are excluded from this document. Figure 3-1 gives the imagery processing flow followed in the enhanced SEDS preprocessor.

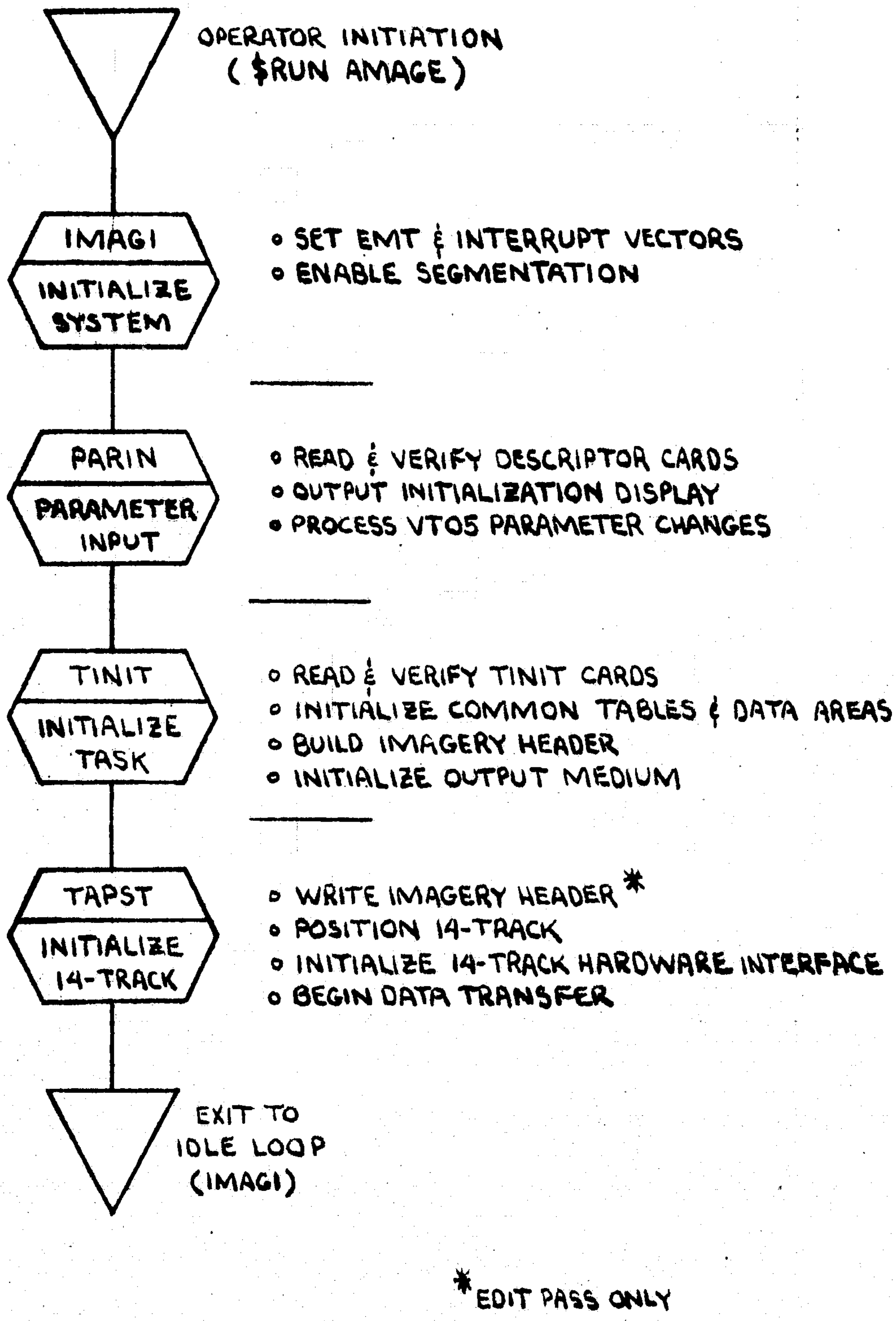


Figure 3-1 Imagery Processing Flow

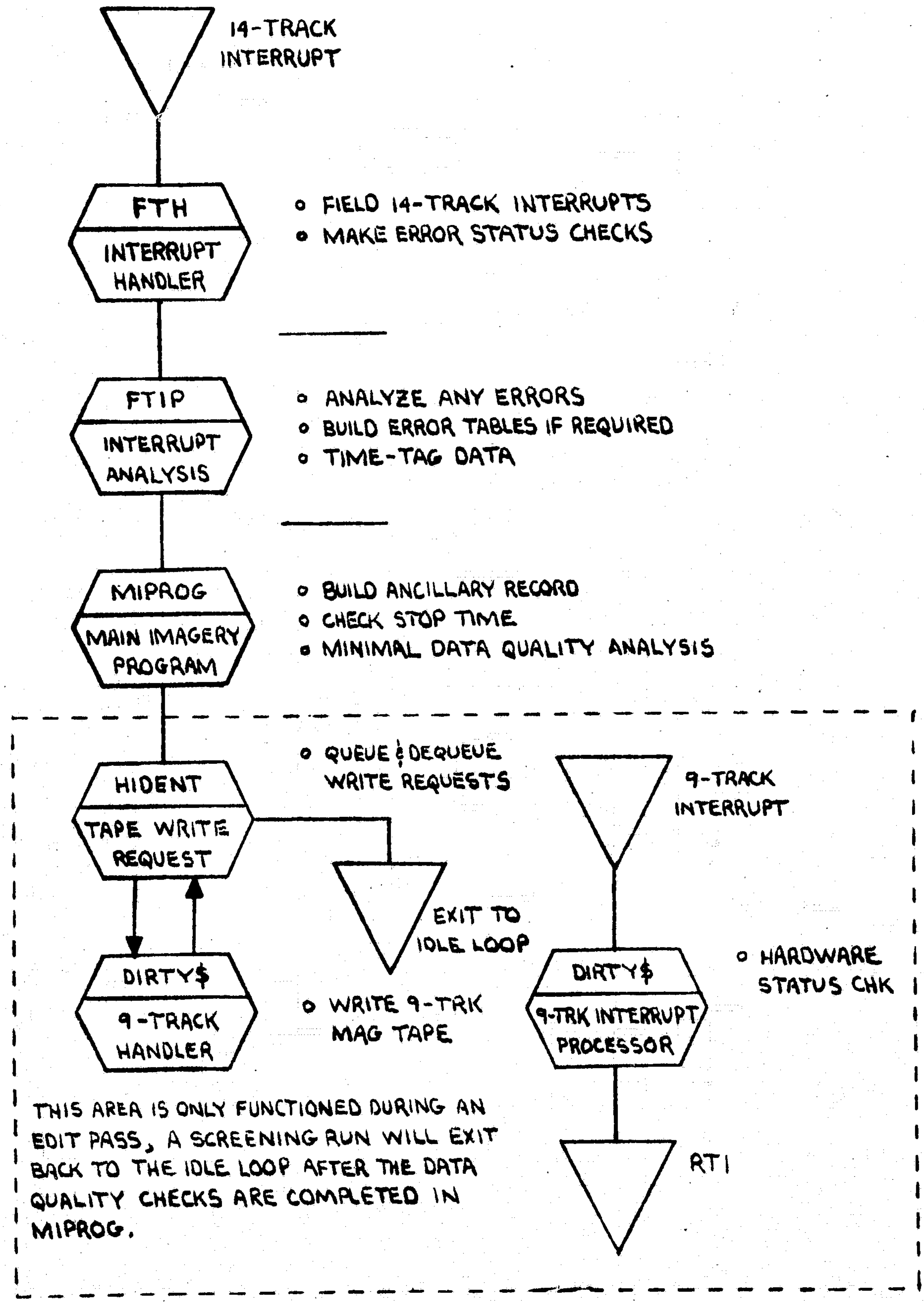


Figure 3-1 (Cont'd)

3.2 INITIALIZATION

The job initialization has been designed to provide the logic required to set up all common tables and data areas, initialize the 14-track tape interface, and begin data transfer, and to direct processing responses for all 14-track interrupts. These tasks are accomplished with the following routines.

A. Task Initialization Routine. Initializes all common data variables and tables as directed by the input job parameters.

B. Tape Start Program. Positions the 14-track tape based on current task parameters, and initializes the track and tape interfaces.

C. 14-Track Interrupt Processor. Initiates proper responses by the data formatting software based on 14-track interrupts.

3.2.1 Task Initialization (TINIT). TINIT is called by PARIN after the initialization parameters have been input and validated. TINIT sets up common tables and data areas for processing as it has been defined by the parameters input from the operator. This routine was enhanced to process and build the tables and constants required to properly process, annotate and format SEDS data. New T14HAN and HEADER default disk files were created to be read by TINIT for SEDS processing (see figures 3-2 and 3-3). The additions made to TINIT deal only with the following specific processing steps.

A. Process the start and stop sample counts as the run requires. These values, for an edit run, must define a scan having an odd number of samples, since 13 bytes of calibration data are added to the scan by the 14-track SEDS logic and the total scan line must terminate on a word boundary. For the screening run, the number of samples in the total scan must be a multiple of 576. This is the pixel width of the screen and is used to compensate for the compression done on the data by the display hardware.

```
$JOB $ED2HD[60,60]
$RU PIP
#SED2HD.MAC<BI:/FA
        .GLOBL  FMTAB
        .TITLE  SEDS.HDR
        ;THIS IS THE HARD CODED HEADER FILE TO BE USED THE THE
        ;PREPROCESSOR WHEN PROCESSING NOAA TAPES FOR SEDS
FMTAB:  .WORD   TEND-TSTART
TSTART:
SEDS.HDR:
        .WORD   154727          ; SYSTEM ID
        .WORD   153705          ; = PREPROCESSOR
        .WORD   153331
        .WORD   142703
        .WORD   161342
        .WORD   154726
        .BLKW   24
        .WORD   153325          ; SENSOR ID
        .WORD   140701          ; = NOAA-3
       .WORD    40362
       .WORD    40100
        .BLKW   17
        .BYTE   10              ; BITS IN PICTUE ELEMENT
        .BYTE   0
        .BYTE   1               ; START ADDR OF VIDEO DATA
        .BYTE   0
        .BYTE   0
        .BYTE   0
        .BYTE   0
        .BYTE   0
        .BYTE   14
        .BYTE   0
        .BYTE   0
        .BYTE   1               ; # CH/PHYSICAL REC
        .BYTE   0
        .BYTE   0
        .BYTE   0
        .BYTE   70.
        .BYTE   0
        .BYTE   0
        .BYTE   0
        .BYTE   0
        .BYTE   0
        .BLKB   200
        .BLKB   200
        .BYTE   1
        .BYTE   1
        .BYTE   1
        .BYTE   1
        .BLKB   174
        .BLKB   200
        .BLKB   100
        .BLKB   100
        .BYTE   0
        .BYTE   0
        .REPT   4
        .BYTE   100
```

Figure 3-2 Header Default Disk File

```
        .BYTE   21
        .ENDR
        .REPT   4
        .BYTE   100
        .BYTE   25
        .ENDR
        .REPT   4
        .BYTE   100
        .BYTE   1
        .ENDR
        .REPT   4
        .BYTE   300
        .BYTE   1
        .ENDR
        .BLKB   2275.
TEND:
        .END
$RU MACRO
*SED2HD,LP:<SED2HD
$RU LINK
*TMK2HD,LP:<TMAKER,SED2HD/E
$FI
```

Figure 3-2 (Cont'd)

ORIGINAL PAGE IS
OF POOR QUALITY

MACRO VR05A 29-AUG-74 00:22 PAGE 1


```
        .TITLE    SEDS.T14
        .GLOBL    FMTAB
        ;FTH COMMAND TABLE USED BY THE 14-TRACK
        ;TAPE HANDLER TO PROCESS NOAA-3 ANALOG
        ;TAPES FOR THE SCREWWORM ERADICATION DATA
        ;SYSTEM
FMTAB:  .WORD     TEND-TSTART
TSTART:
SEDS.T14:
        .WORD     0                     ;BIAS TO NEXT TASK TABLE
        .BYTE     3                     ;CCNTR
        .BYTE     1                     ;INITF
        .BYTE     0                     ;SUBCODE
        .BYTE     0                     ;CODE
        .BYTE     0                     ;S/A
        .BYTE     0                     ;DENSITY
        .BYTE     74                    ;TAPE SPEED INTEGER
        .BYTE     0                     ;TAPE SPEED FRACTION
        .WORD     0                     ;DATA SOURCE/TKX/TYPE CODE/BN
        .BYTE     74                    ;RECORDED SPEED INTEGER
        .BYTE     0                     ;RECORDED SPEED FRACTION
        .BYTE     0                     ;RECORDED DATA RATE INTEGER
        .BYTE     0                     ;RECORDED DATA RATE FRACTION
        .WORD     0                     ;DATA SOURCE/TKX/TYPE CODE/BN
        .BYTE     0                     ;RECORDED DATA RATE INTGER
        .BYTE     0                     ;RECORDED DATA RATE FRACTION
        .WORD     0                     ;TFA
        .WORD     0                     ;IRIG TRANSLATOR COMMAND
        .WORD     2000                  ;TIIR
        .WORD     100214                ;PI
        .WORD     0                     ;PARAMETER
        .BYTE     1                     ;SUBCODE
        .BYTE     0                     ;CODE
        .WORD     32                    ;BIAS TO GET TICB
        .BYTE     1                     ;E/SUBCODE
        .BYTE     1                     ;CODE
        .WORD     0                     ;TAPE FAULT ADDRESS
        .BYTE     1                     ;FPBUF
        .BYTE     1                     ;FPERO
        .BYTE     7                     ;BITS/WDS
        .BYTE     0                     ;WRDS/SF
        .WORD     0                     ;NOT USED
        .WORD     0                     ;NOT USED
        .WORD     0                     ;NOT USED
        .RAD50    /SED/                 ;DESCRIPTOR FILE NAME
        .RAD50    /S/                   ;DESCRIPTOR FILE NAME
        .RAD50    /HUR/                 ;EXTENSION
        .WORD     13                    ;CFN
        .WORD     0                     ;BIAS TO NEXT BLOCK
        .BYTE     0                     ;TFC
        .BYTE     13                    ;NOT ACTIVE/TRACK #
        .WORD     0                     ;PST
        .BYTE     0                     ;PSTI
        .BYTE     24                    ;PSTL
        .WORD     154002                ;D/A,PACK,CH 1/2 ACTIVE,TCR,A/D RES
        .WORD     60100                 ;TFWA1
        .WORD     0                     ;TFWA2
```

Figure 3-3 FTH Command Table (T14HAN)

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
MACRO VR05A 29-AUG-74 00:22 PAGE 1+

        .WORD   60100                   ;TFWA3
        .WORD   0                       ;TFWA4
        .WORD   1                       ;TSYNC1 OR START PIXEL #
        .WORD   6654                    ;TSYNC2 OR STOP PIXEL #
        .WORD   0                       ;TMAXK1
        .WORD   0                       ;TMASK2
        .WORD   0                       ;WORDS/FRAME
        .WORD   003400                  ;SYNC WIN,SYNC ERR CT,BITS/WD,FRAMES/BUF
TEND:
        .END
```


Figure 3-3 (Cont'd)

B. Calculate the words per frame based on the start/stop sample counts. This value is derived by adding 13 to the difference of the stop sample minus the start sample, doubling the sum, and subtracting 1 (see figure 2-2 in section 2).

C. Incorporate the SEDS parameters into existing common tables. The tables affected are the 14-track parameter table (T14HAN), the header record table (HEADER) and various formatting tables holding variables defining tape output format.

D. Build a table to be used in initializing the 525-line color display (IDPBA; see figure 3-4). This table is set up to reflect the status of the input parameters describing the input tape. The table will ultimately define to the hardware in what manner the data is to be displayed. The table is totally constructed here in TINIT, but is not transferred to the hardware status registers until TAPST gains control.

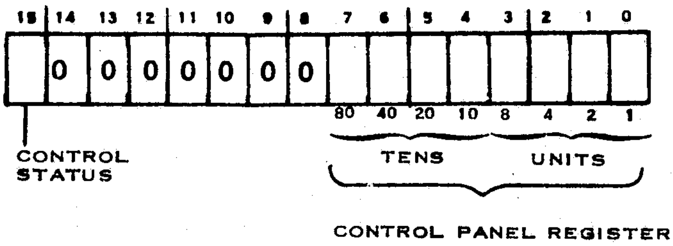
Figure 3-5 gives a detailed flow of the additional code inserted for SEDS preprocessor implementation.

3.2.2 Tape Start Program (TAPST). TAPST positions the 14-track tape based on the current task start parameters, and initializes the tape and track interfaces from the tables constructed in TINIT. The modifications installed in TAPST for SEDS implementation impact only the following areas.

A. The code to write the 9-track universal header is conditionally branched around if processing is for a SEDS screening pass.

B. For SEDS processing, the time search routine in TAPST was modified to make use of the time code translator installed for the SEDS task. The same logic flow is followed, except that new filter weights are used and communication is accomplished through a different hardware control register. Figure 3-6 gives the structure for this new command word.

IDPBA (STATUS)

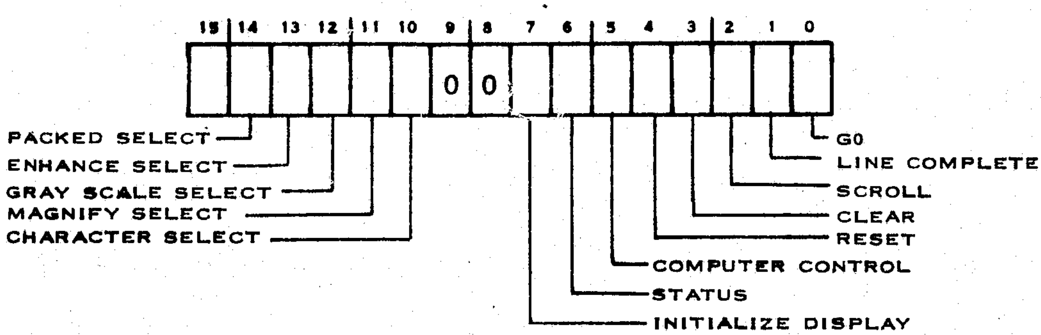


Used only to return status to the user program.

Bit 15 - If set, indicates that format of displayed image is selected from display operator's control panel; if cleared, indicates that format control is being selected by software.

Bits 7-0 - Two BCD digits dialed in at the panel with values of 00-99; used for options and controlled by the operator.

IDPBA + 2 (FORMAT AND COMMAND)



Bit 15 - If set, the computer control command must be executed before the format can be software-selected.

Bits 14-10 instruct the handler to select the specified display format.

Bit 14 - If set, refresh data is displayed with color assignments following: (LSB & MSB) Bits 2 and 3 = blue; Bits 4 and 5 = green; Bits 6 and 7 = red. This bit may not be used with enhance selected (see bit 13).

Bit 13 - If set, data will be modified by contents of enhancement look-up tables before being displayed.

Figure 3-4 Imagery Display Parameter Block (IDPBA)

Bit 12 - If set, a gray scale or color bar will be displayed across lower 48 lines of displayed image.

Bit 11 - If set, area bracketed by the cursor will be magnified to fill the screen.

Bit 10 - If set, pixel value will be converted to an alphanumeric character and displayed.

Bits 7-0 command the display handler to initiate the described action.

Bit 7 - This command executes in a single call a reset and clear on the display and positions the cursor at the home position (see figure 2-2).

Bit 6 - This command causes no interaction with the display, but simply returns to the parameter block the current display status, control register value and X-Y cursor position.

Bit 5 - Reverts display format control to the user software.

Bit 4 - Initializes display logic that keeps count of even/odd scans for automatic scrolling.

Bit 3 - Erases the refresh memory.

Bit 2 - Scrolls the image in the selected direction, up or down. If scroll mode is selected in the input parameter word (see below), the image will scroll automatically after every other GO.

Bit 1 - Signifies the completion of a scan line that has been constructed one color at a time.

Bit 0 - Indicates a message transfer.

IDPBA + 4 (INPUT PARAMETERS)

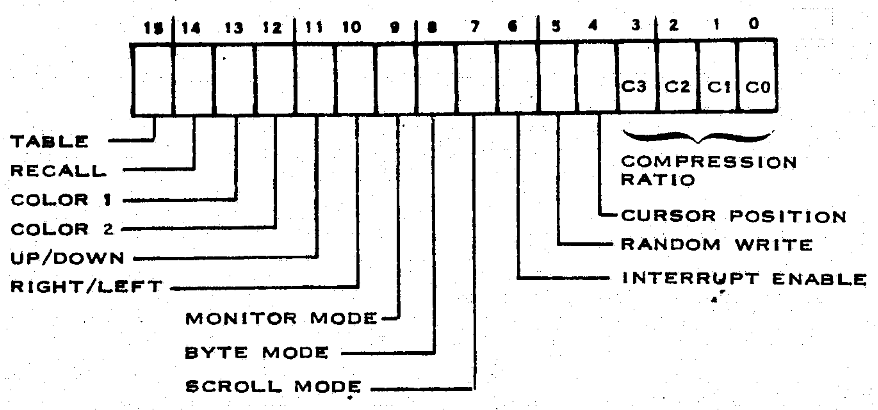


Figure 3-4 (cont'd)

Expands on the type of data transfer that is to take place, specifies the purpose of the data and identifies the mode of operation of the display.

Bit 15 - Specifies that this message transfer is to one of three enhancement tables (one for each color). Desired table is selected by bits 13 and 12. The format of data for enhancement table must be specific, with the upper byte containing the address in the table to receive the data and the lower byte containing the data value itself. This is accomplished by setting bit 12 in the Format and Command Word and reading a scan line from the color bar which is displayed.

Bit 14 - Specifies that this message transfer is from the display memory addressed by line and segment.

Bits 13-12 - In combination, these color tag each message to or from the display unit as follows: 0-1 = red with LSB/MSB 6/7; 1-0 = green with LSB/MSB 4/5; 1-1 = blue with LSB/MSB 2/3; 0-0 = black and white (all 6 bits) with LSB/MSB 2/7.

Bit 11 - If set, displayed image will move up by two scan lines when a scroll command or condition occurs; if cleared, image will move down.

Bit 10 - If set, causes update message to be displayed left-to-right; if cleared, data will be displayed right-to-left.

Bit 9 - Allows display unit to trap data from the UNIBUS for screening; only one call to IDEH is required to display an image in this mode. UNIBUS is monitored by the display hardware for the final word address (IDPBA + 10), then begins trapping data for subsequent addresses until the message length (IDPBA + 12) is satisfied. When it is, the display again monitors UNIBUS for final word address. If SCROLL MODE is selected, image will scroll in the desired direction after every other scan.

Bit 8 - Causes display unit to discard high order byte of each update message transfer.

Bit 7 - Allows display to scroll after every other message transfer to the display; i.e., after every other GO.

Bit 6 - Permits display unit to generate interrupts.

Bit 5 - Specifies that associated message transfer is to a random line and/or segment on the display; bit is required to keep the update from affecting the scroll and compression logic.

Figure 3-4 (cont'd)

Bit 4 - In this call only, positions the cursor to the point specified by the X and Y positions passed in the parameter block, IDPBA 14 and 16.

Bits 3-0 - Applied to every data message sent to the display unit; first pixel will be the *Cth* byte and every *Cth* byte following the first displayed pixel.

IDPBA + 6 (READ/WRITE LINE AND SEGMENT)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L9	L8	L7	L6	L5	L4	L3	L2	L1	L0	S5	S4	S3	S2	S1	S0
LINE NUMBER										SEGMENT NUMBER					

Allows user to address display image as to where a message is output to or recalled from (see figure 3-).

Bits 15-6 - Specify image line number to be updated or recalled. For scrolling bottom to top, use line = 0; for scrolling top to bottom, use line = 429.

Bits 5-0 - Specify image starting segment number to be updated or recalled from. Each line is divided into 19 segments of 32 pixels each, with only 18 segments visible. The 19th scan line in blanking can be updated and recalled, so that it can be used for such things as scan line identification.

IDPBA + 10 (FIRST WORD ADDRESS)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0

Specifies the 16 least significant bits of the update or recall buffer start address.

IDPBA + 12 (MEMORY EXTENSION BITS AND MESSAGE LENGTH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A 17	A 16	M 13	M 12	M 11	M 10	M 9	M 8	M 7	M 6	M 5	M 4	M 3	M 2	M 1	M 0
MEX BITS		LINE NUMBER													

Figure 3-4 (cont'd)

Bits 15-14 - Specify the two most significant bits of the update or recall buffer First Word Address (see above). These bits allow buffer addresses of 18 bits.

Bits 13-0 - Specify the number of 16-bit word transfers in the update or recall message.

IDPBA + 14 (X-POSITION)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

Specifies the binary count horizontal location of the center of the cursor with respect to the displayed image (see figure 2-2). Home position = 1140 (608_{10}). The cursor is two segments (64 pixels) wide.

IDPBA + 16 (Y-POSITION)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

Specifies the binary count vertical location of the center of the cursor with respect to the displayed image (see figure 2-2). Home position = 030 (24_{10}). The cursor is 48 lines high.

Figure 3-4 (cont'd)

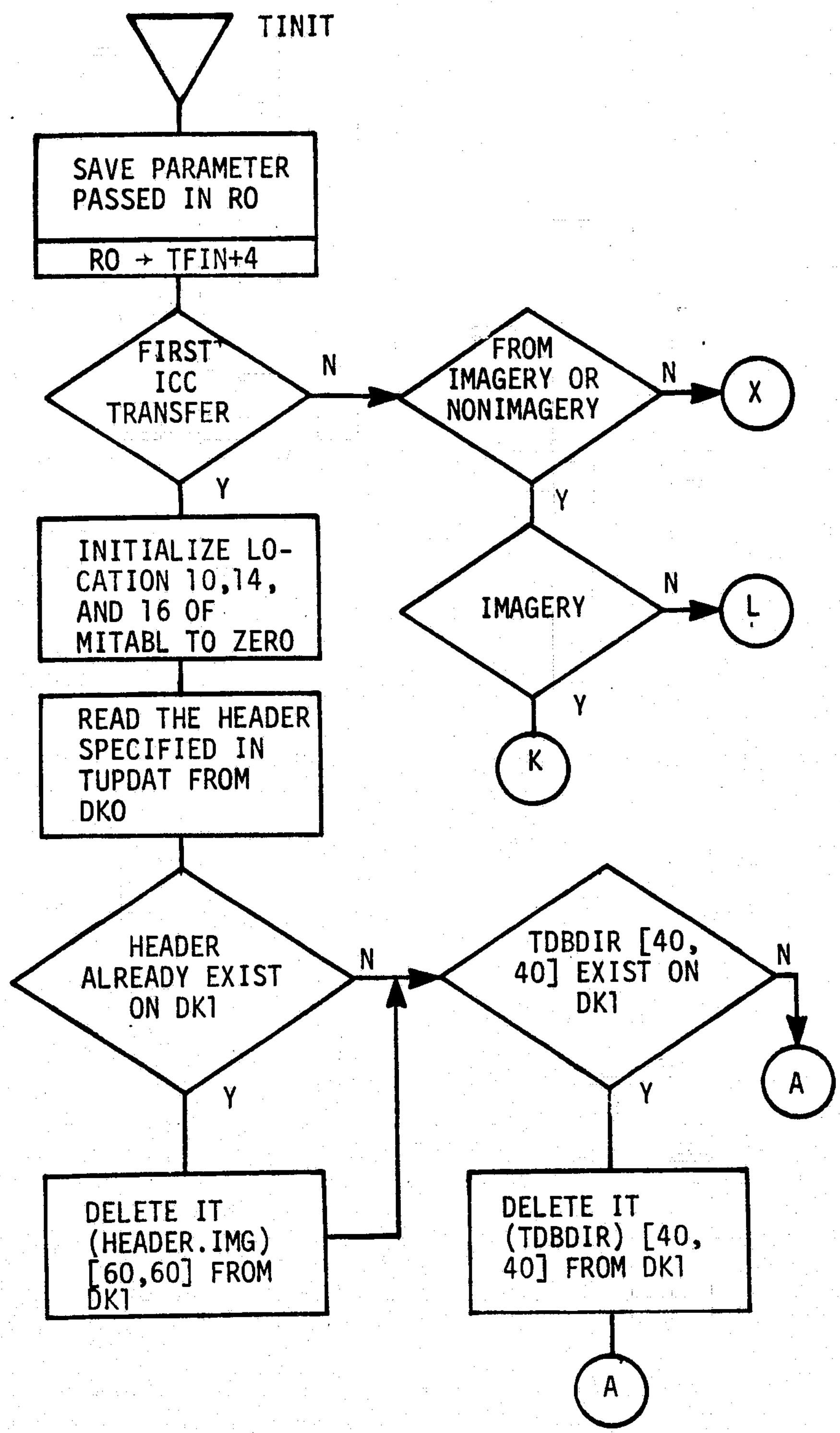


Figure 3-5 Flow of Code Insertions for SEDS Preprocessor Implementation (1 of 30)

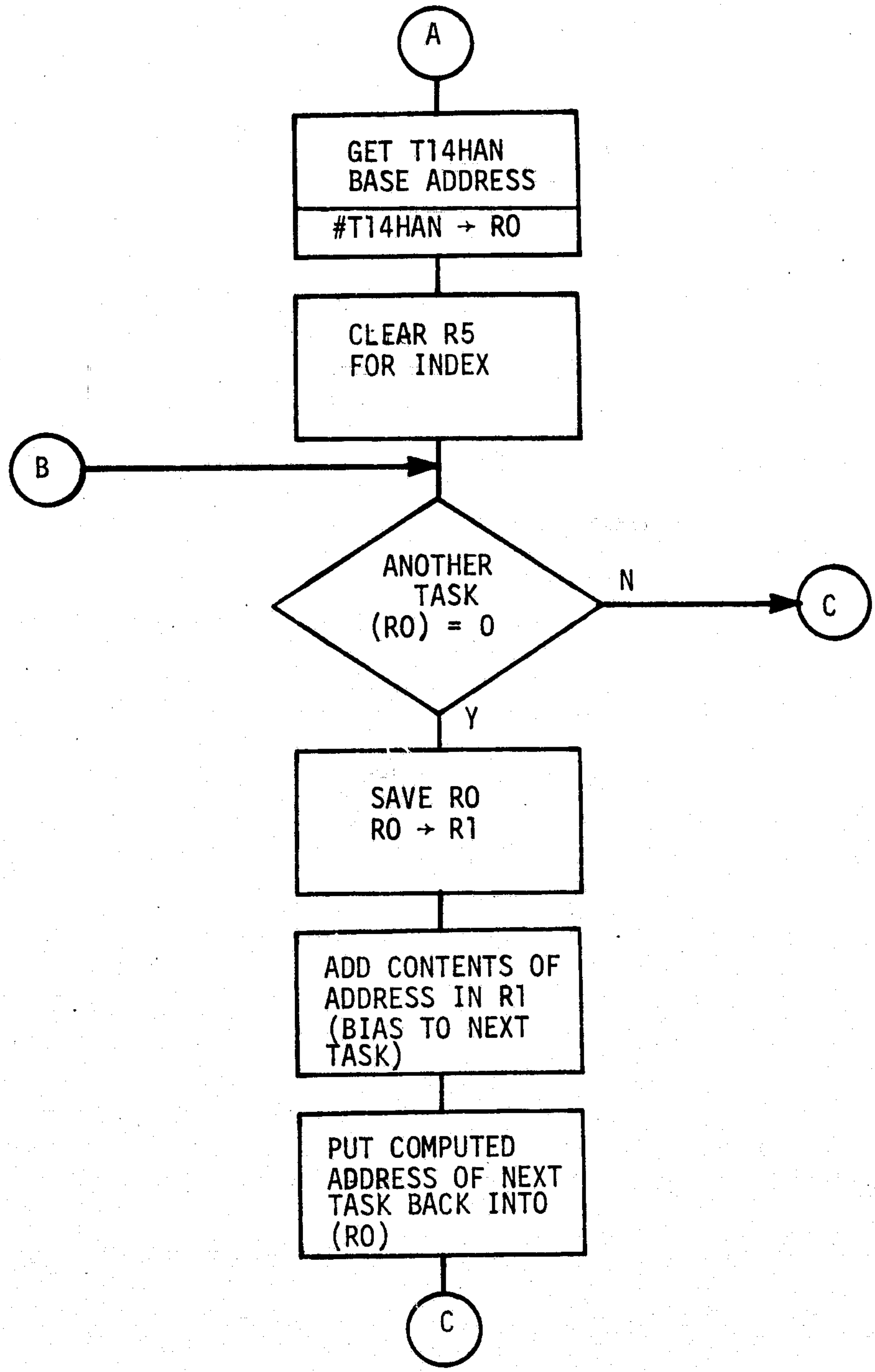


Figure 3-5 (2 of 30)

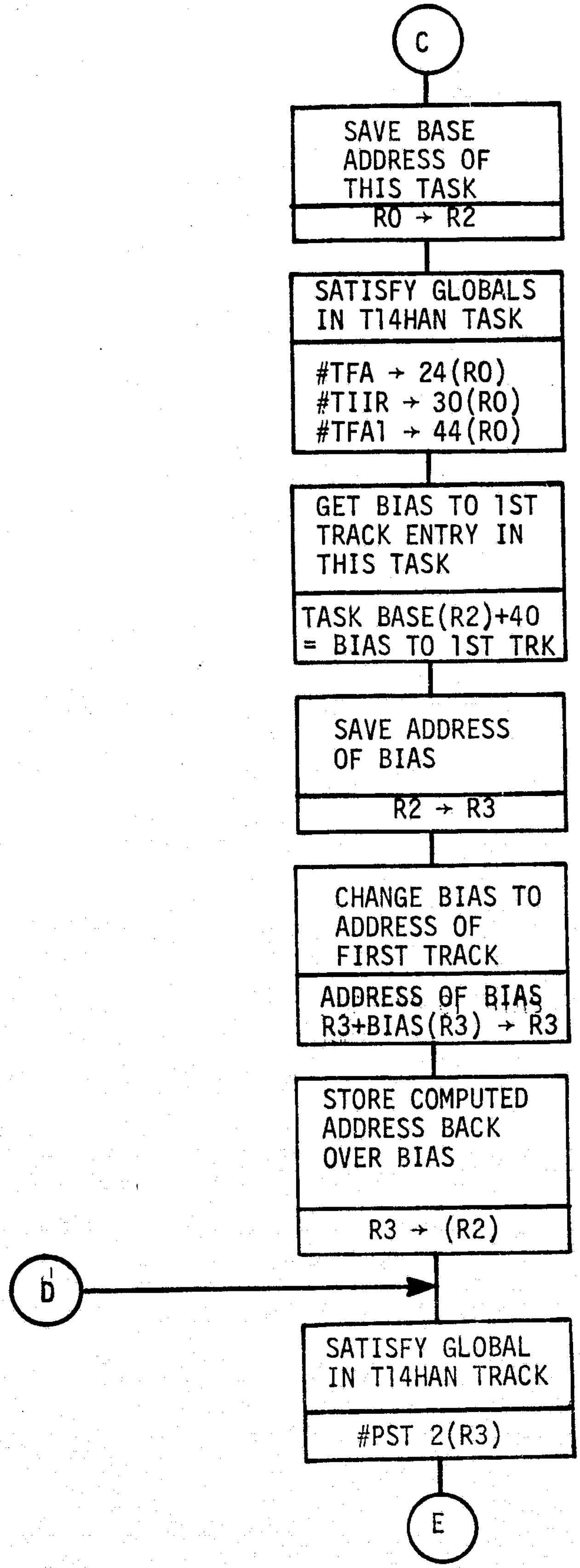


Figure 3-5 (3 of 30)

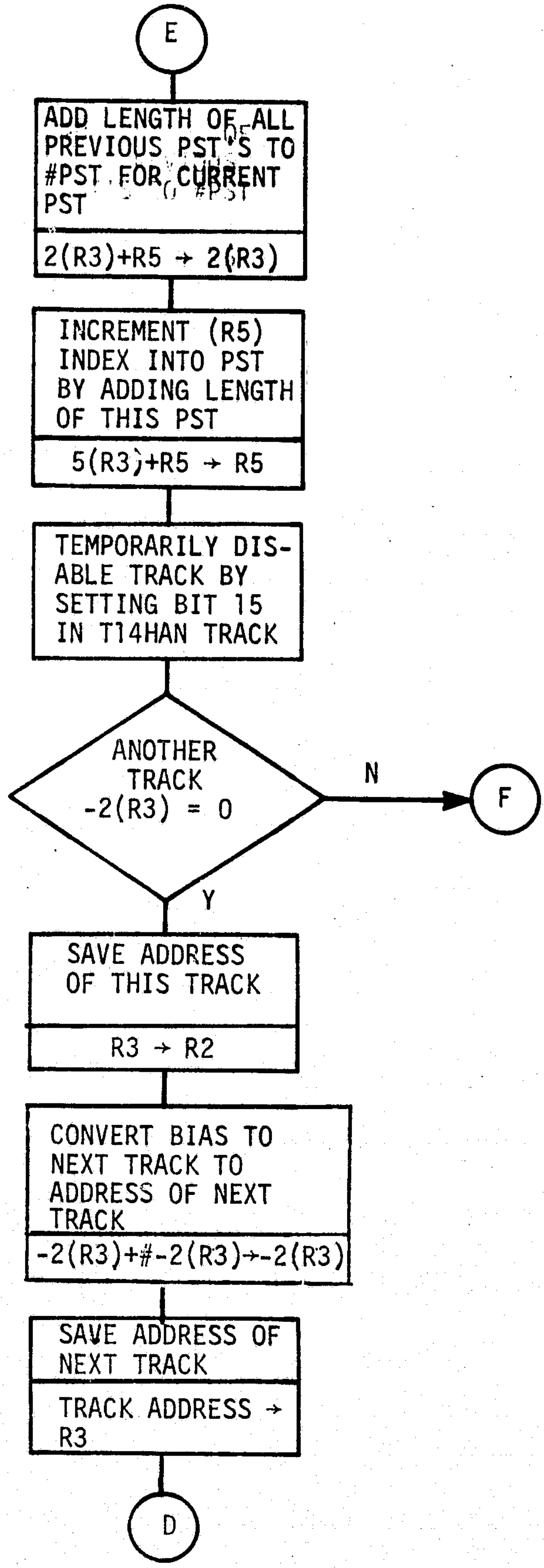


Figure 3-5 (4 of 30)

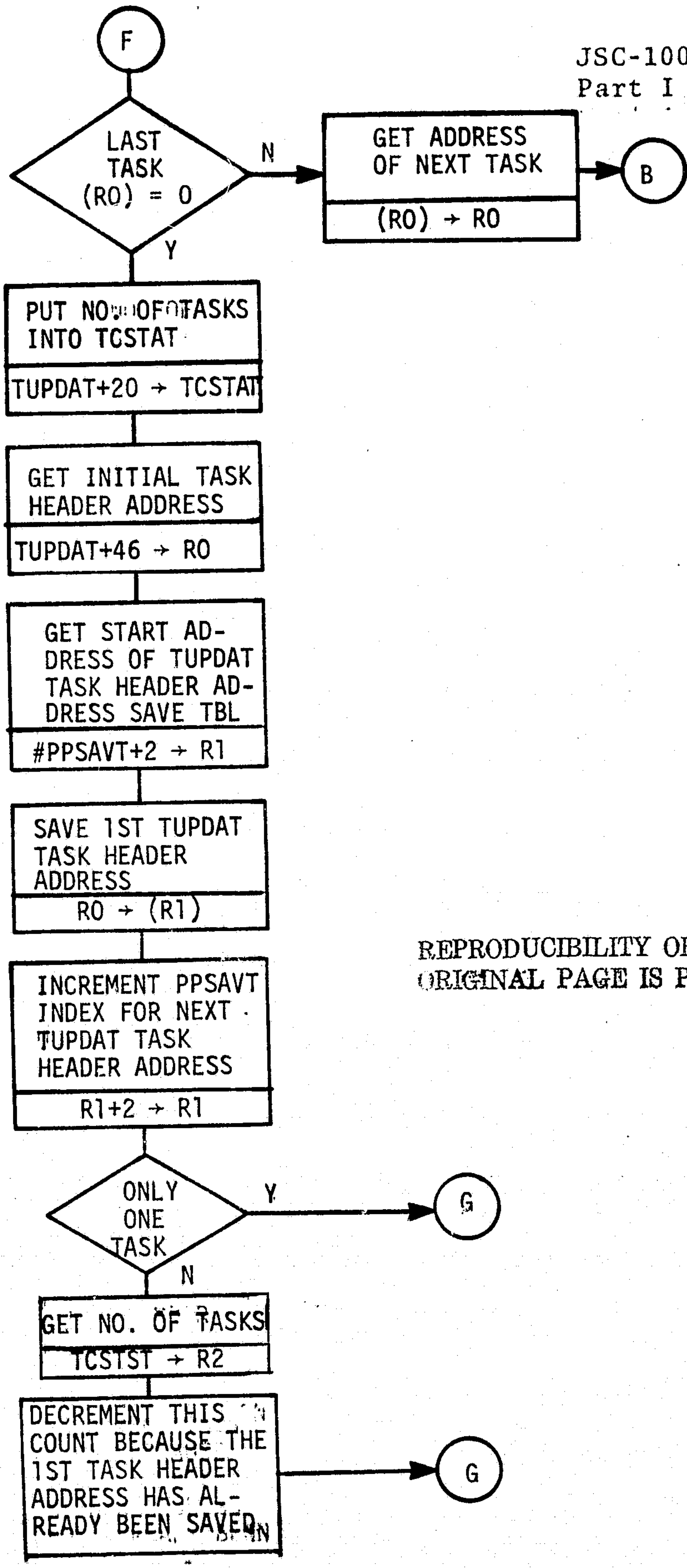


Figure 3-5 (5 of 30)

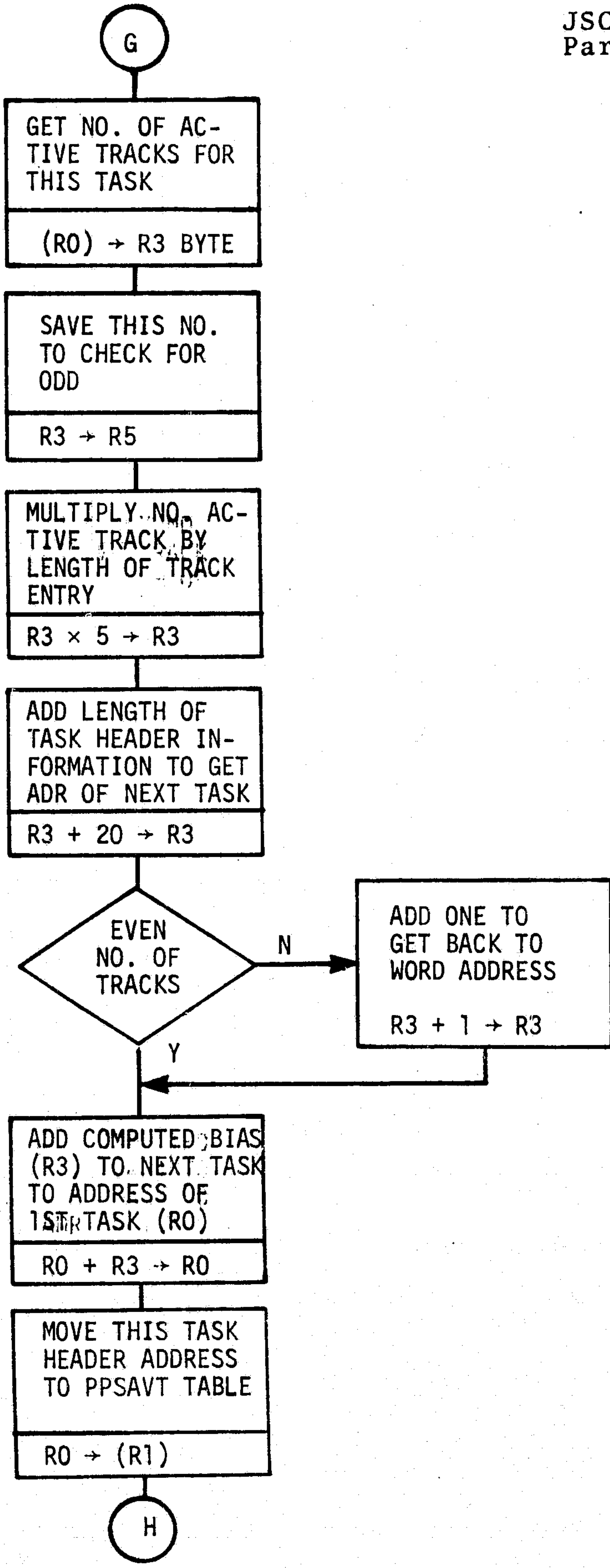


Figure 3-5 (6 of 30)

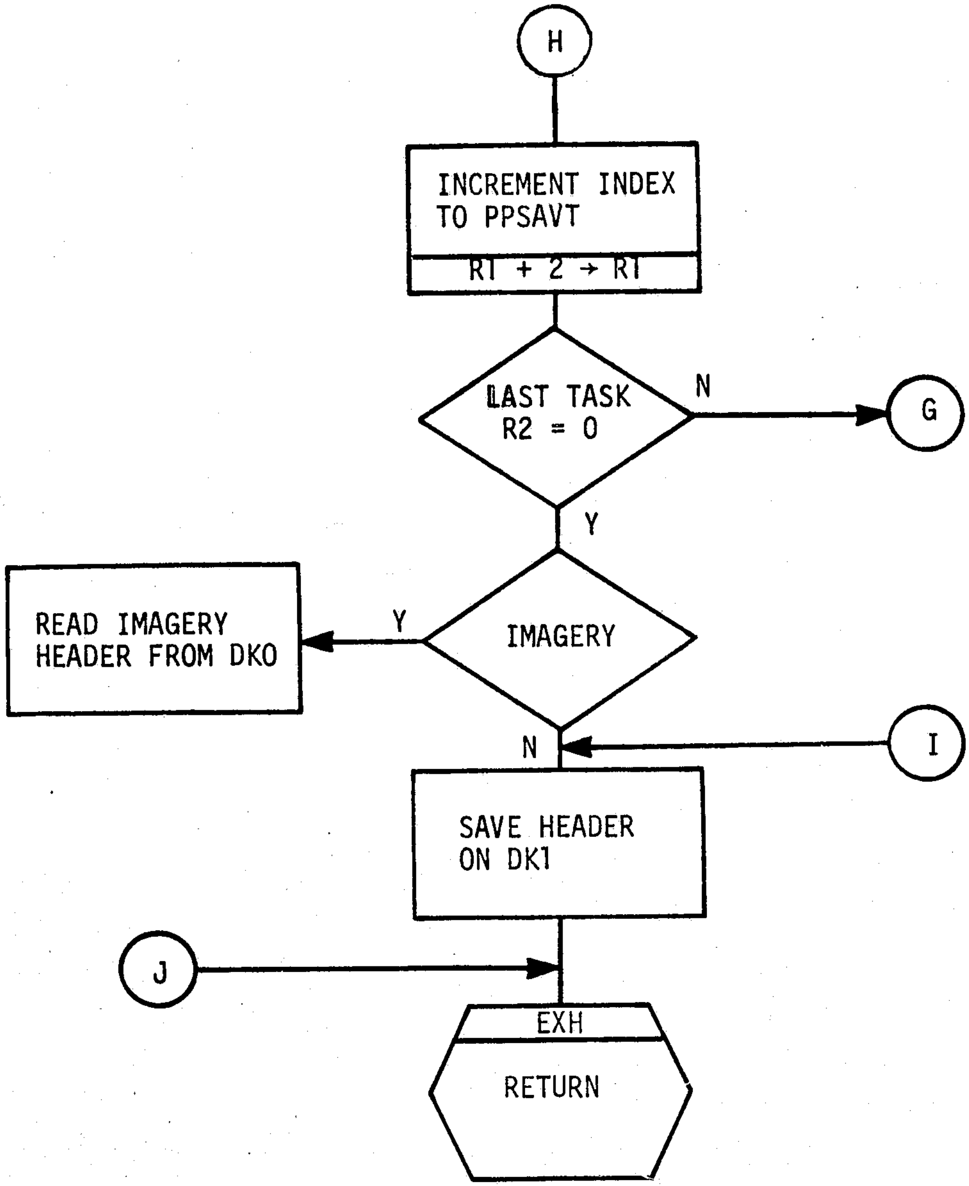


Figure 3-5 (7 of 30)

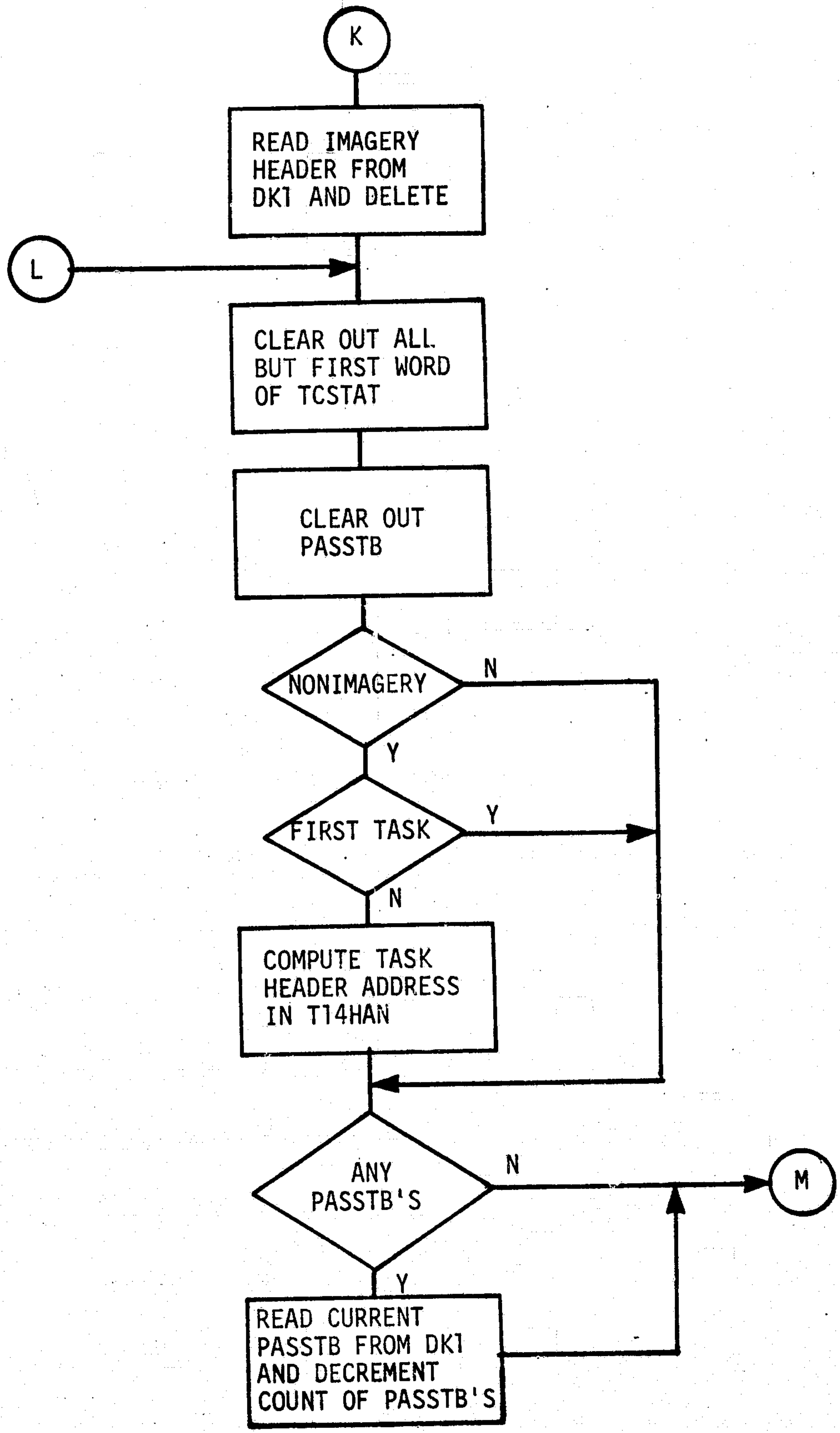


Figure 3-5 (8 of 30)

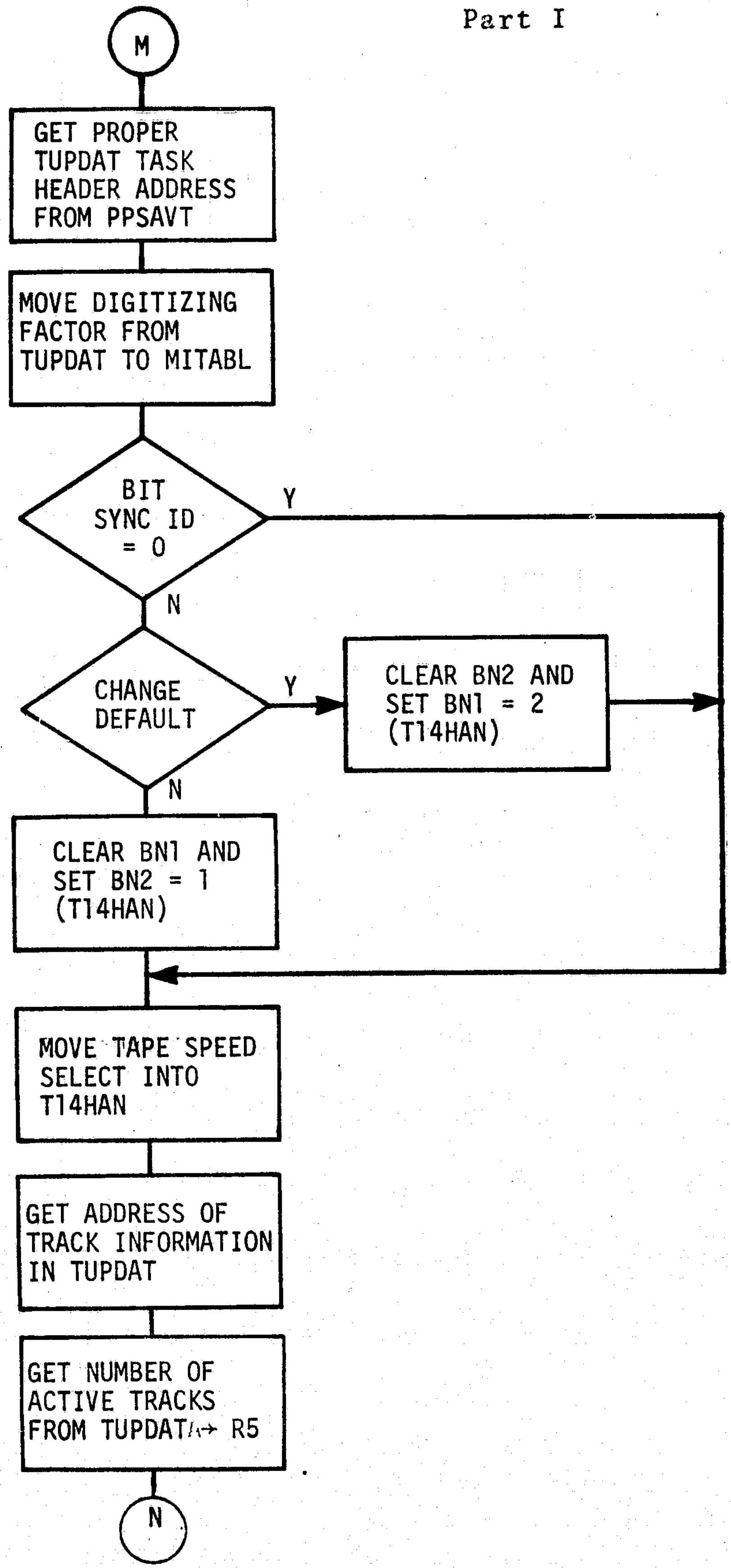


Figure 3-5 (9 of 30)

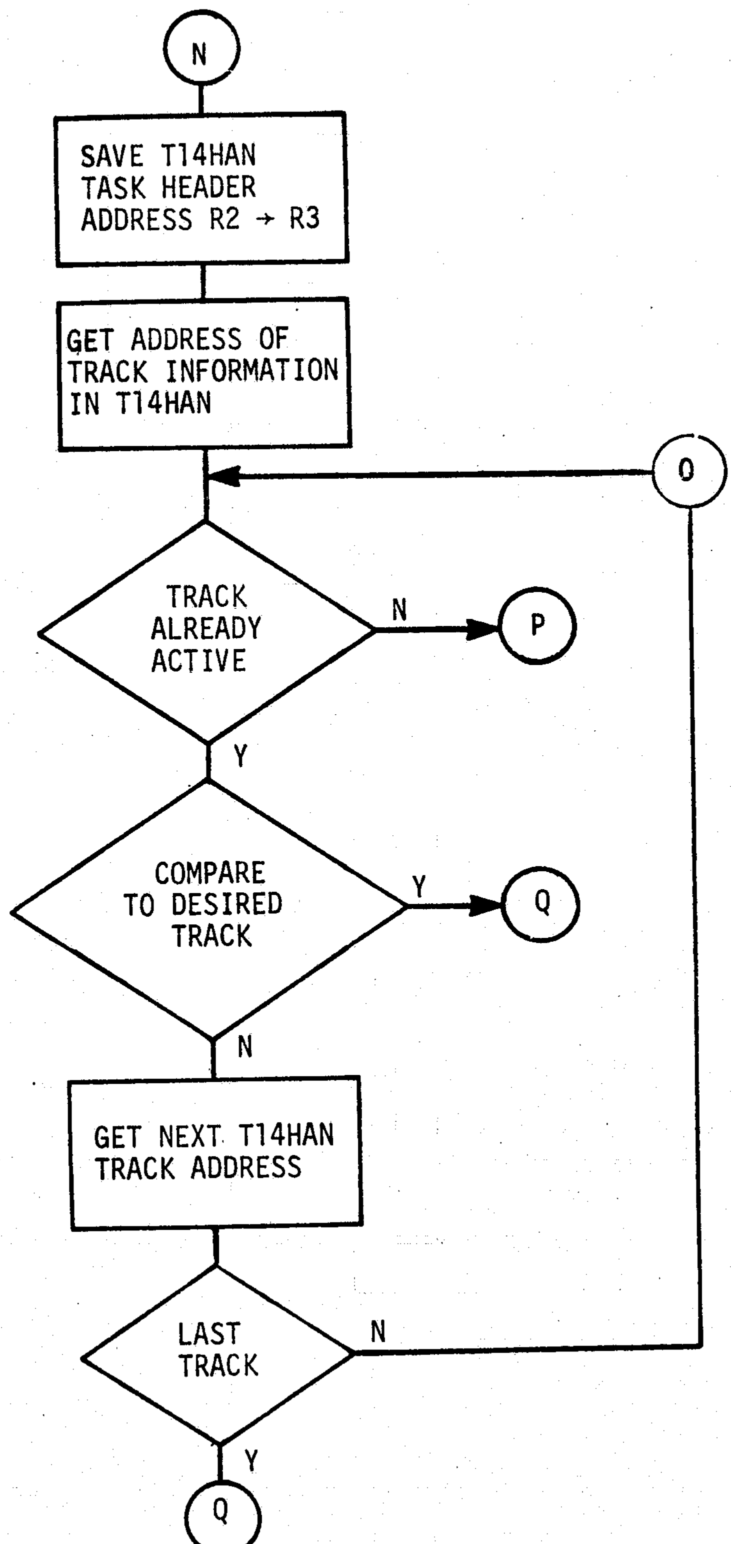


Figure 3-5 (10 of 30)

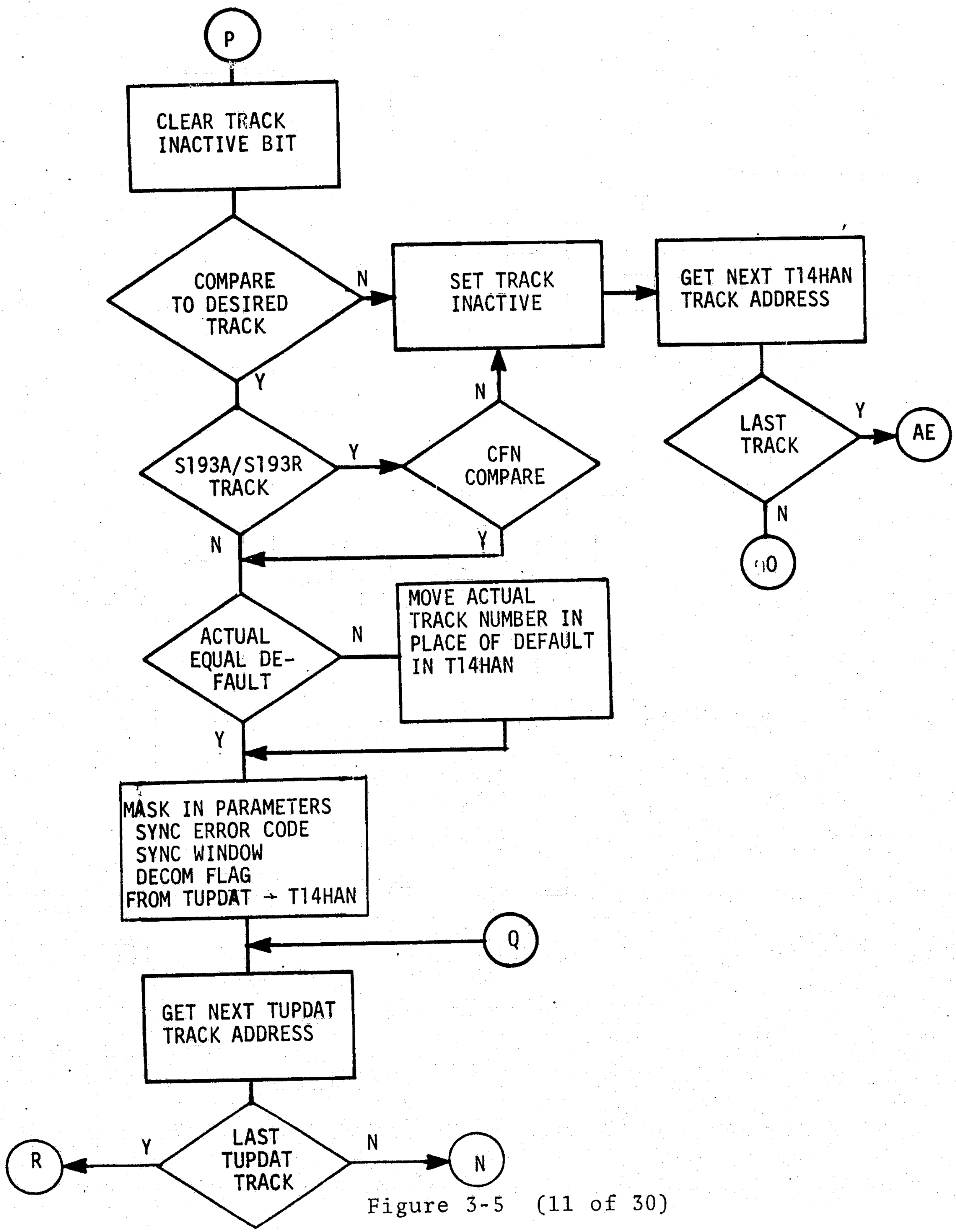


Figure 3-5 (11 of 30)

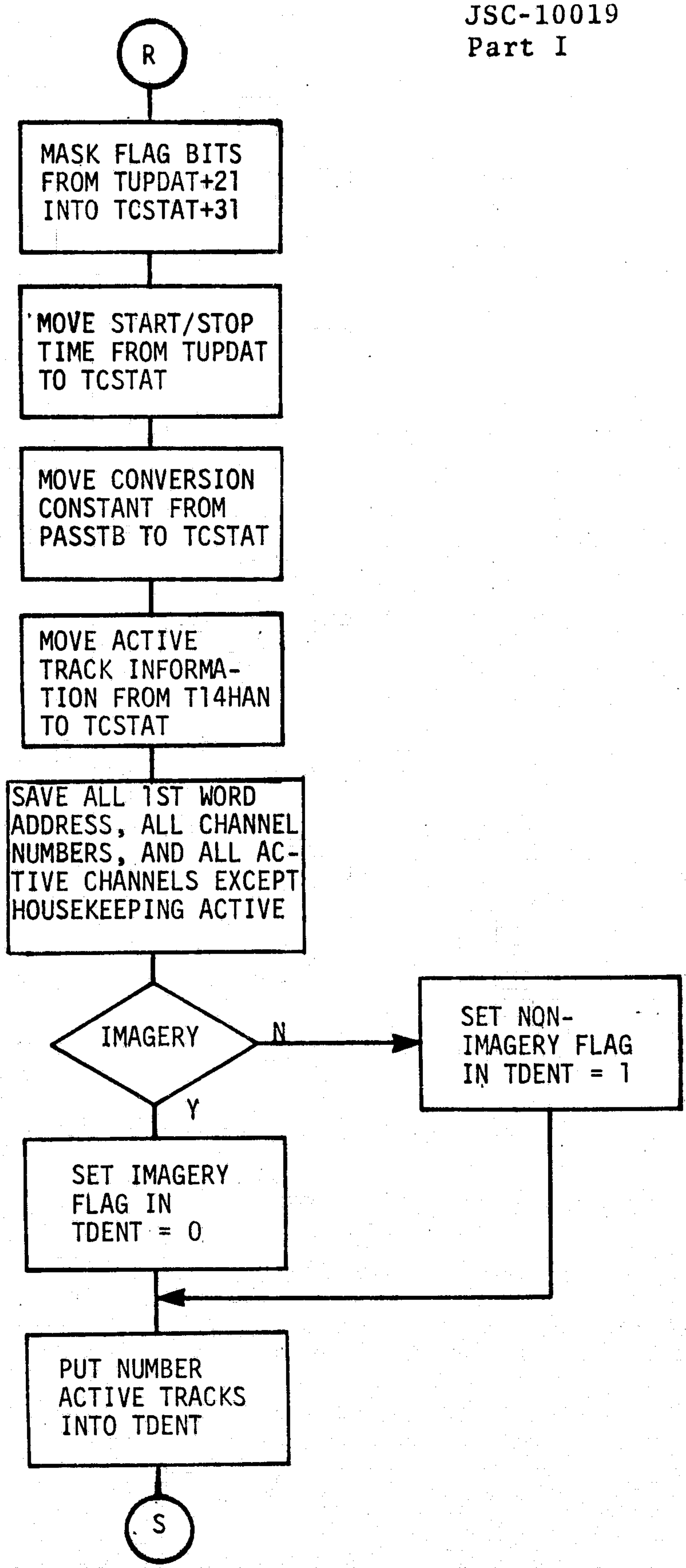


Figure 3-5 (12 of 30)

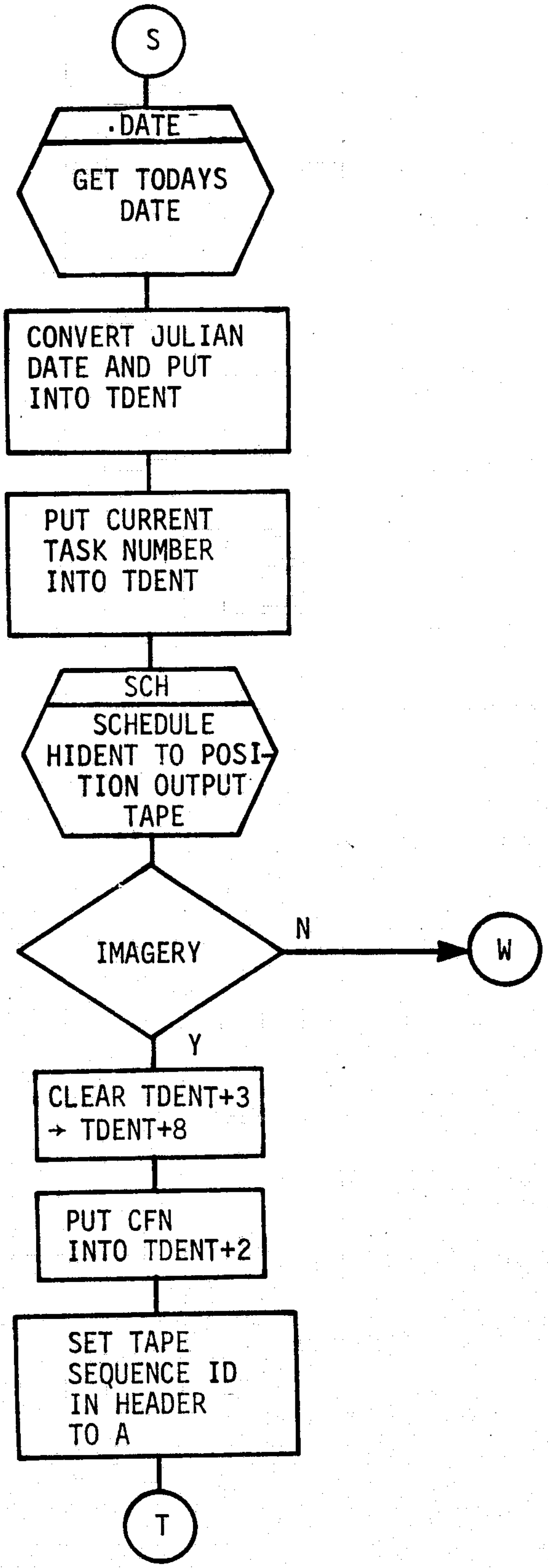


Figure 3-5 (13 of 30)

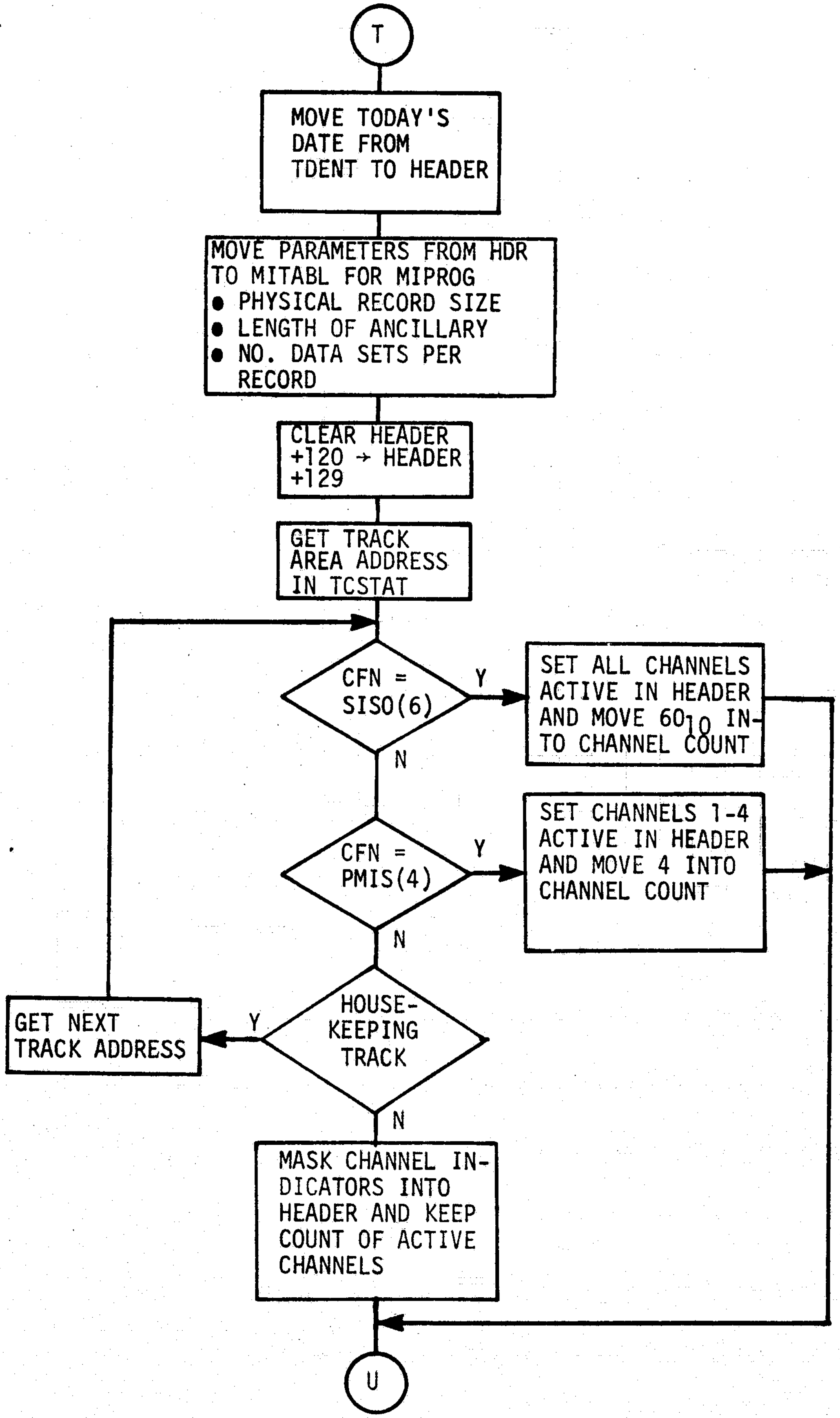


Figure 3-5 (14 of 30)

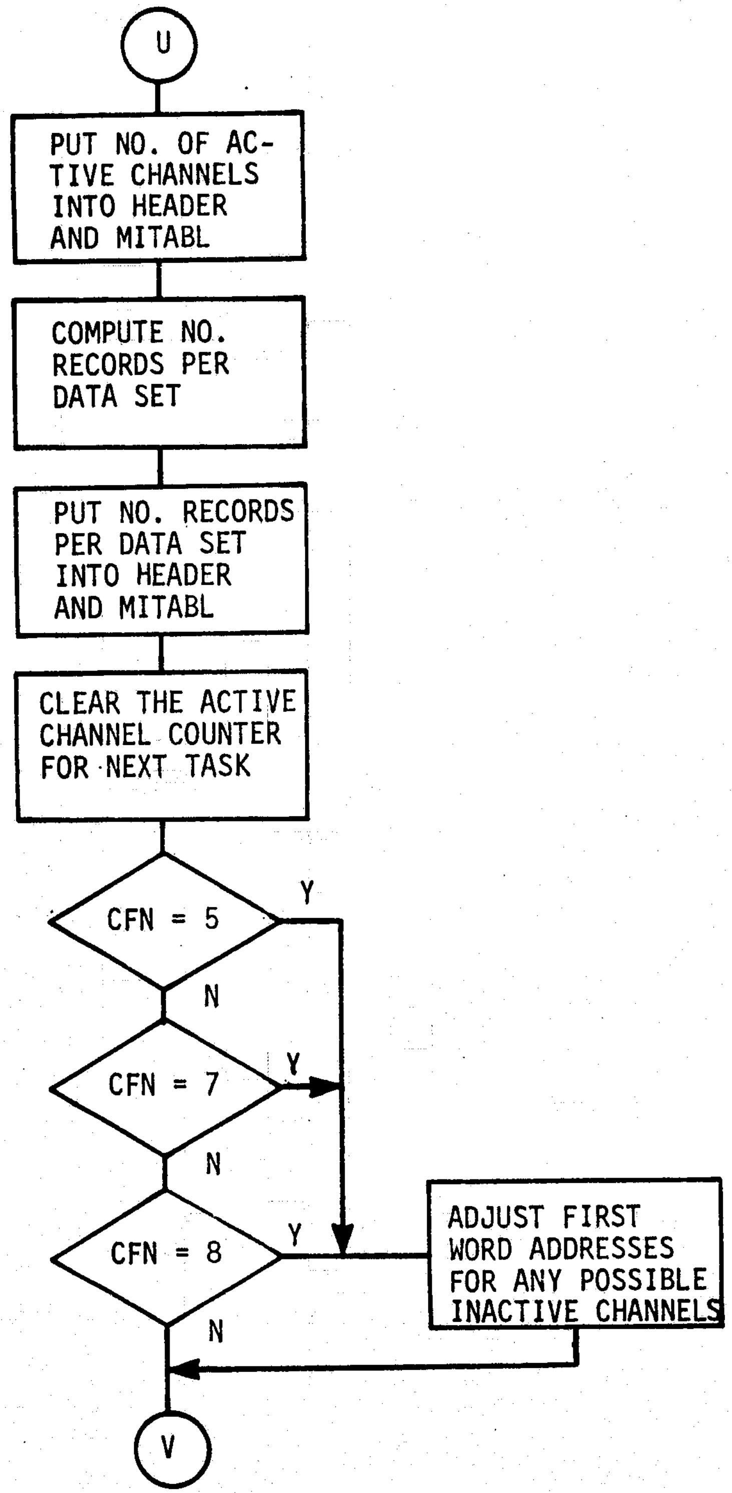


Figure 3-5 (15 of 30)

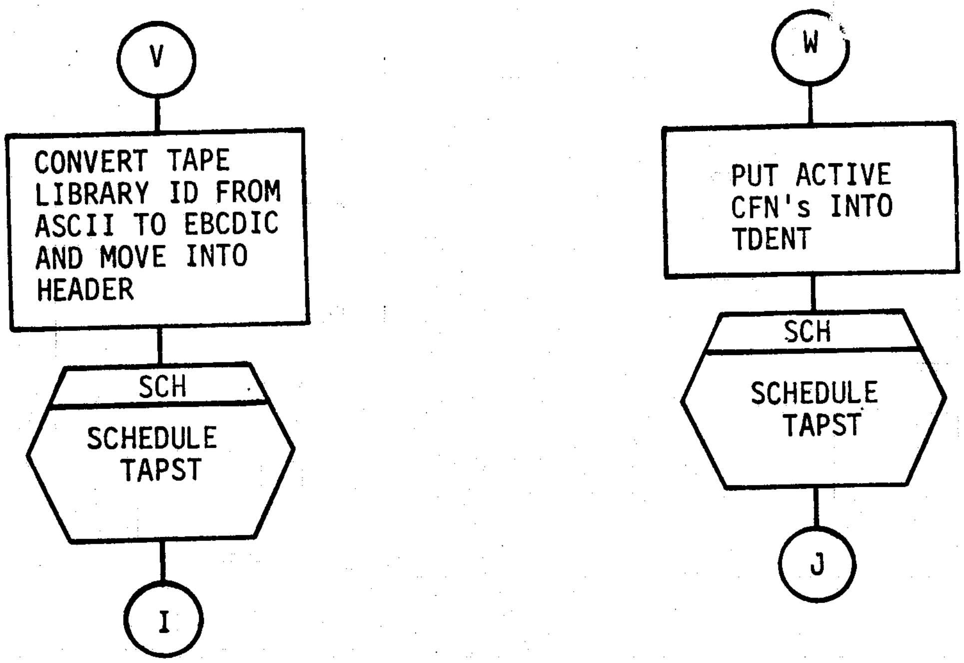


Figure 3-5 (16 of 30)

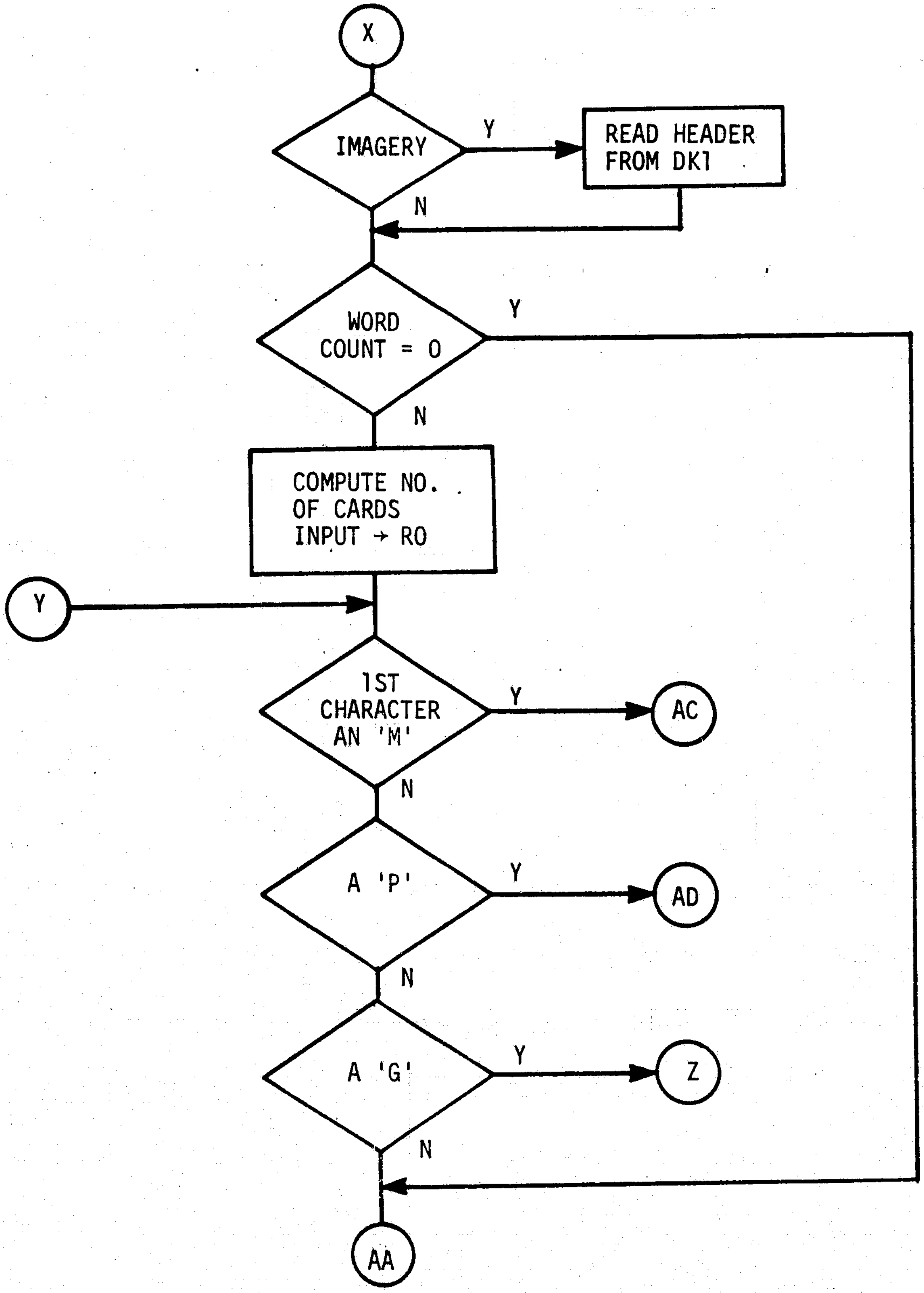


Figure 3-5 (17 of 30)

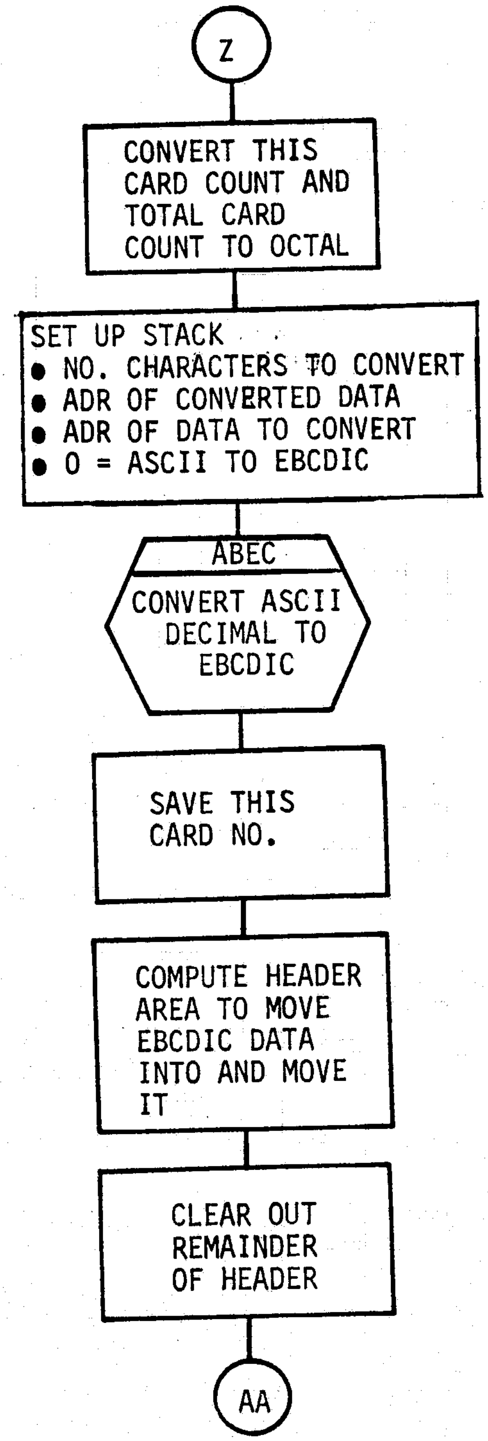


Figure 3-5 (18 of 30)

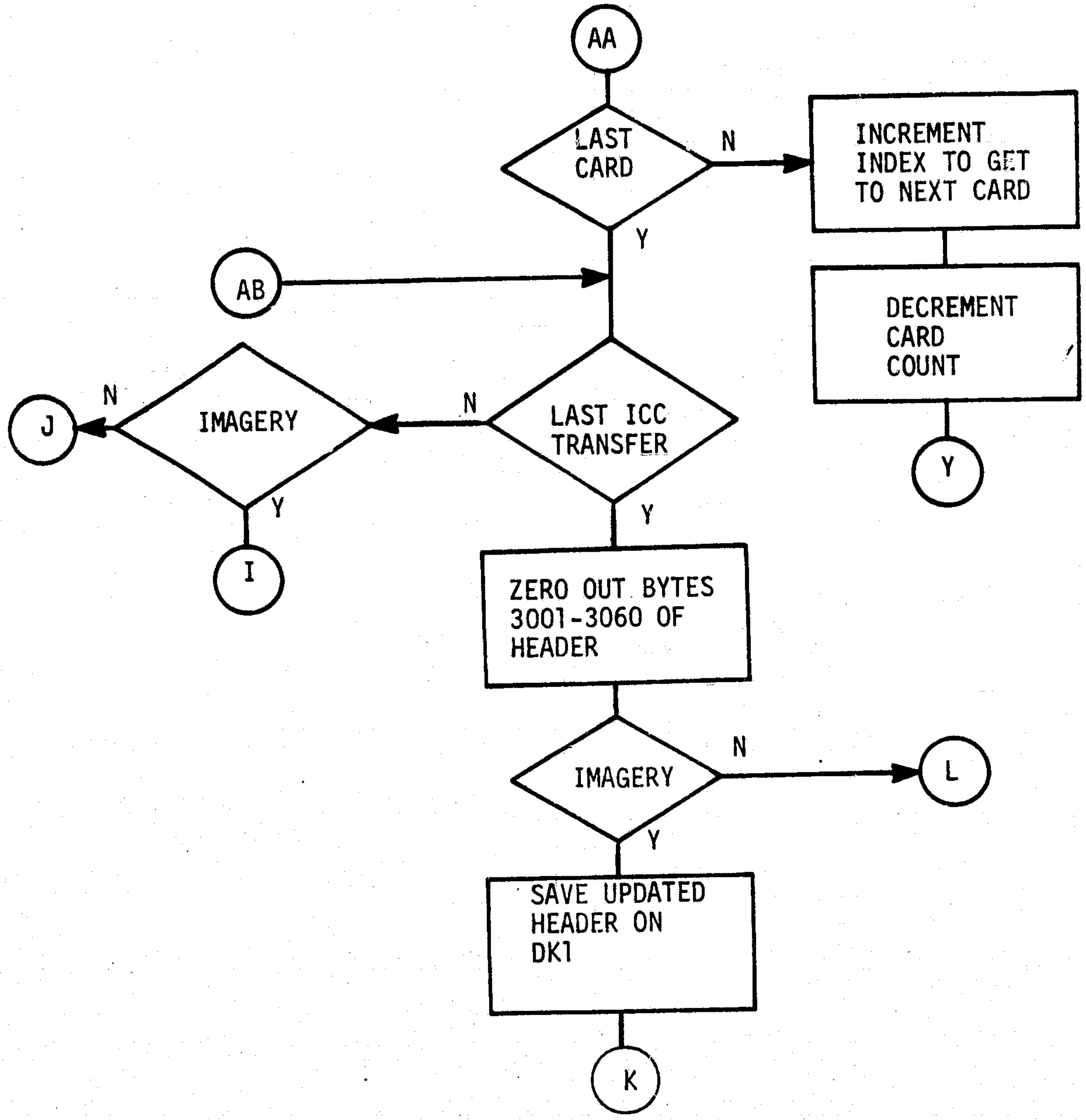


Figure 3-5 (19 of 30)

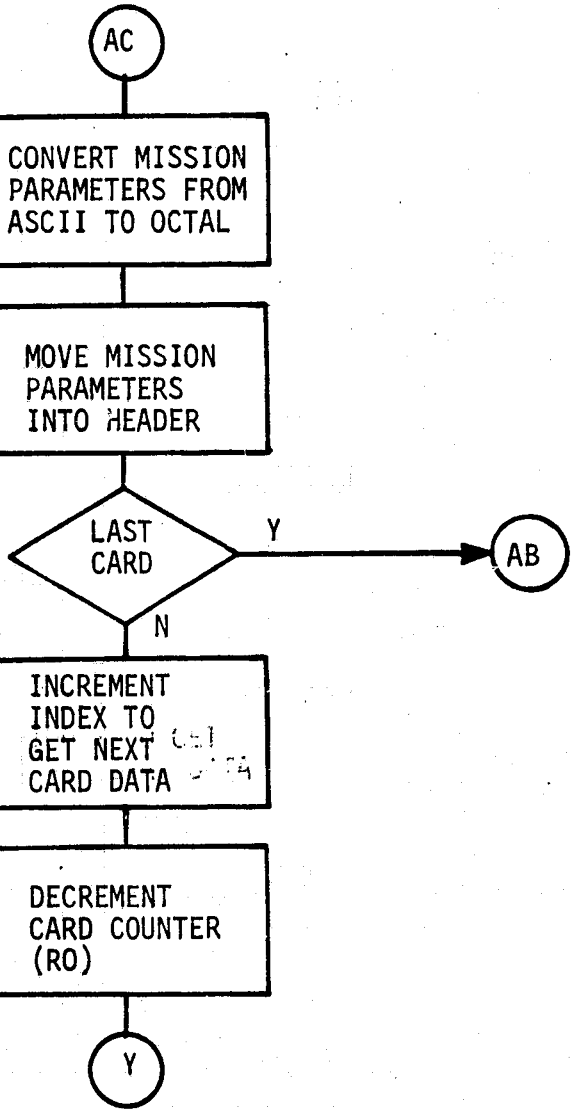


Figure 3-5 (20 of 30)

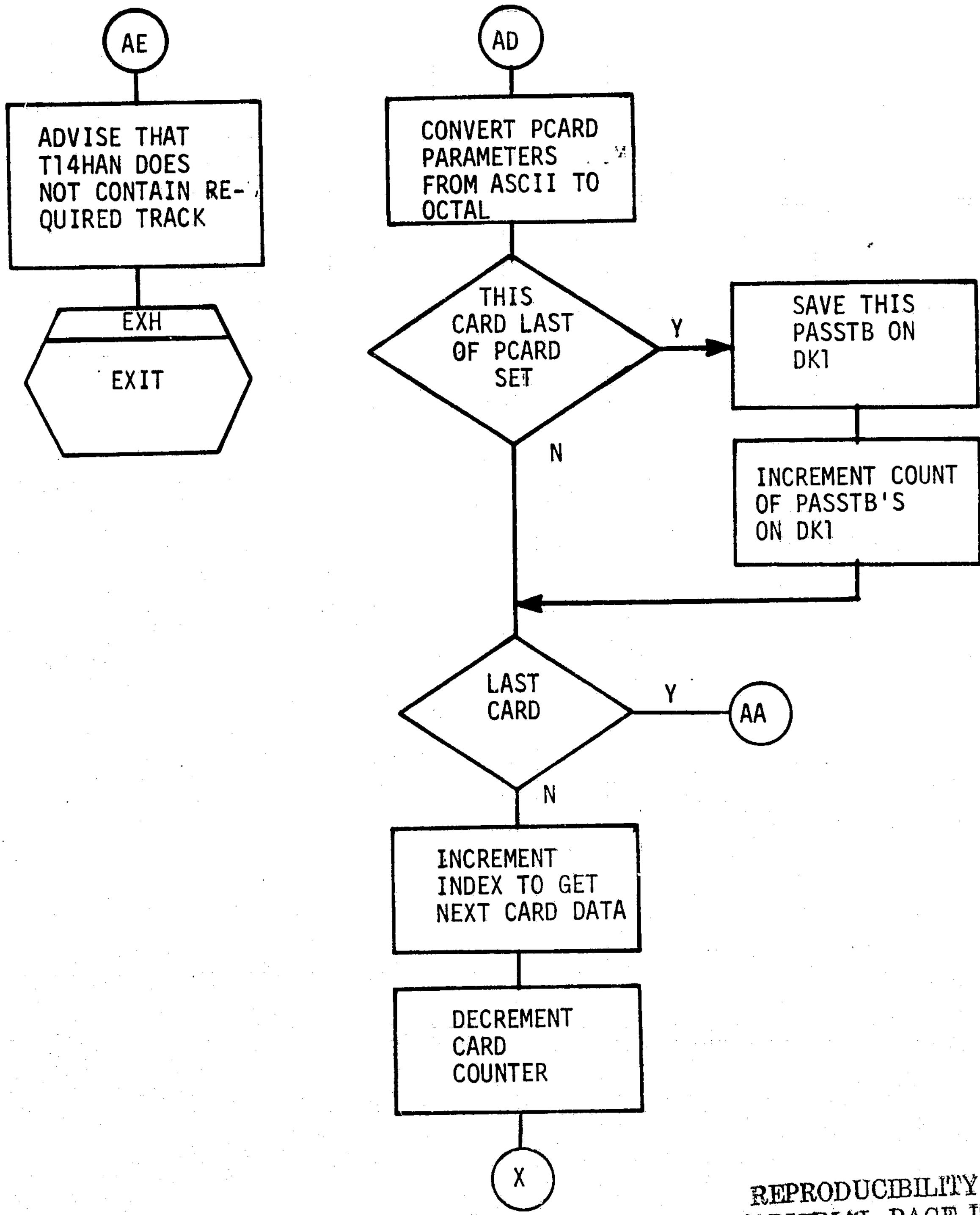


Figure 3-5 (21 of 30)

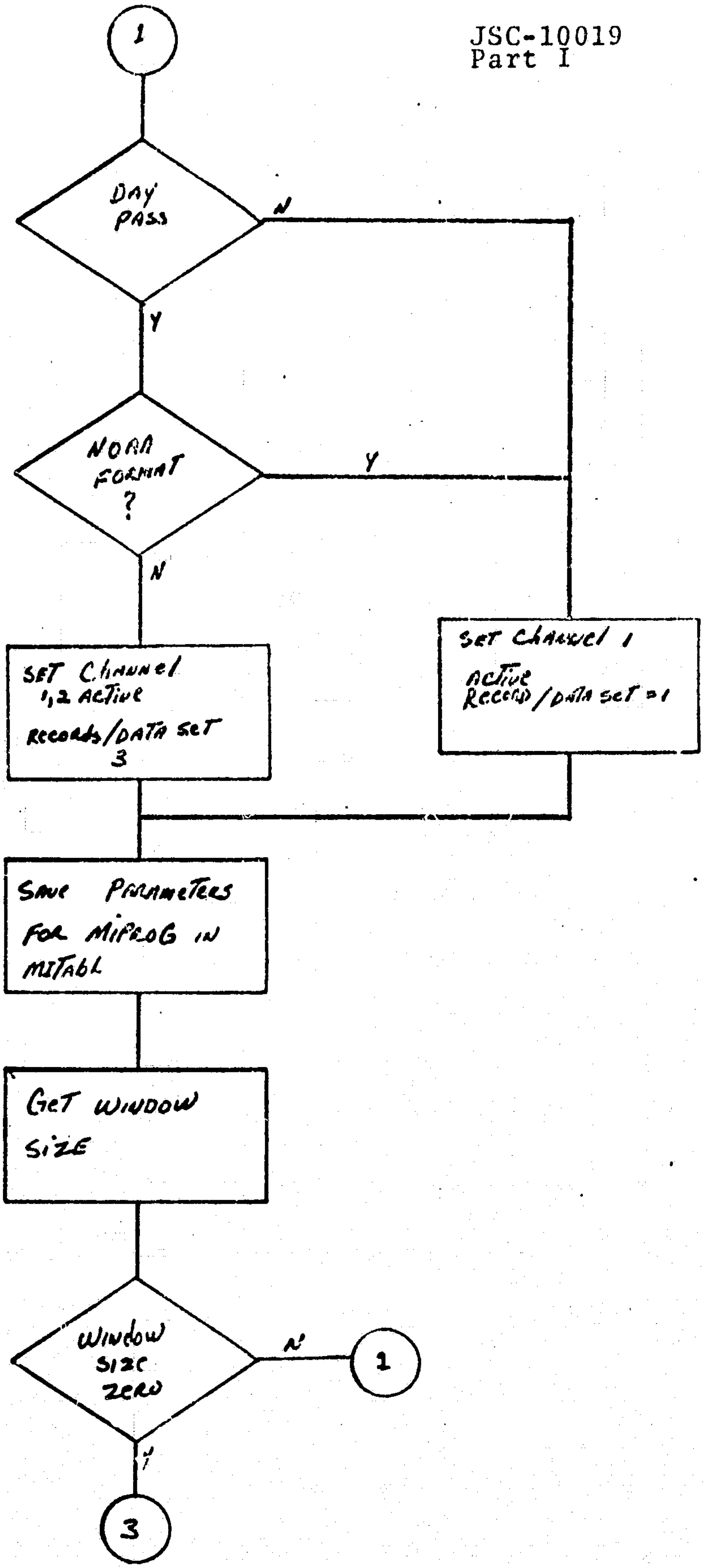


Figure 3-5 (22 of 30)

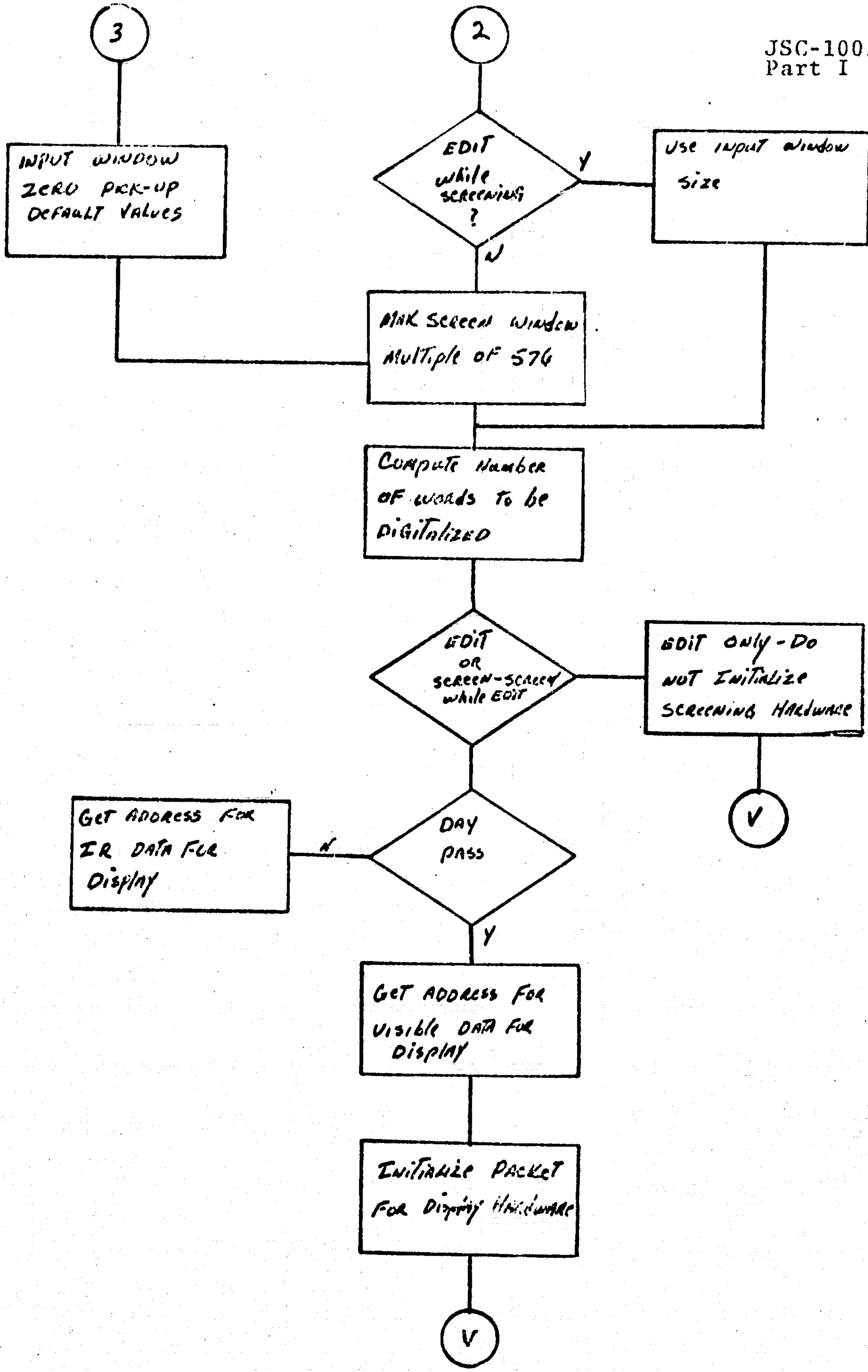


Figure 3-5 (23 of 30)

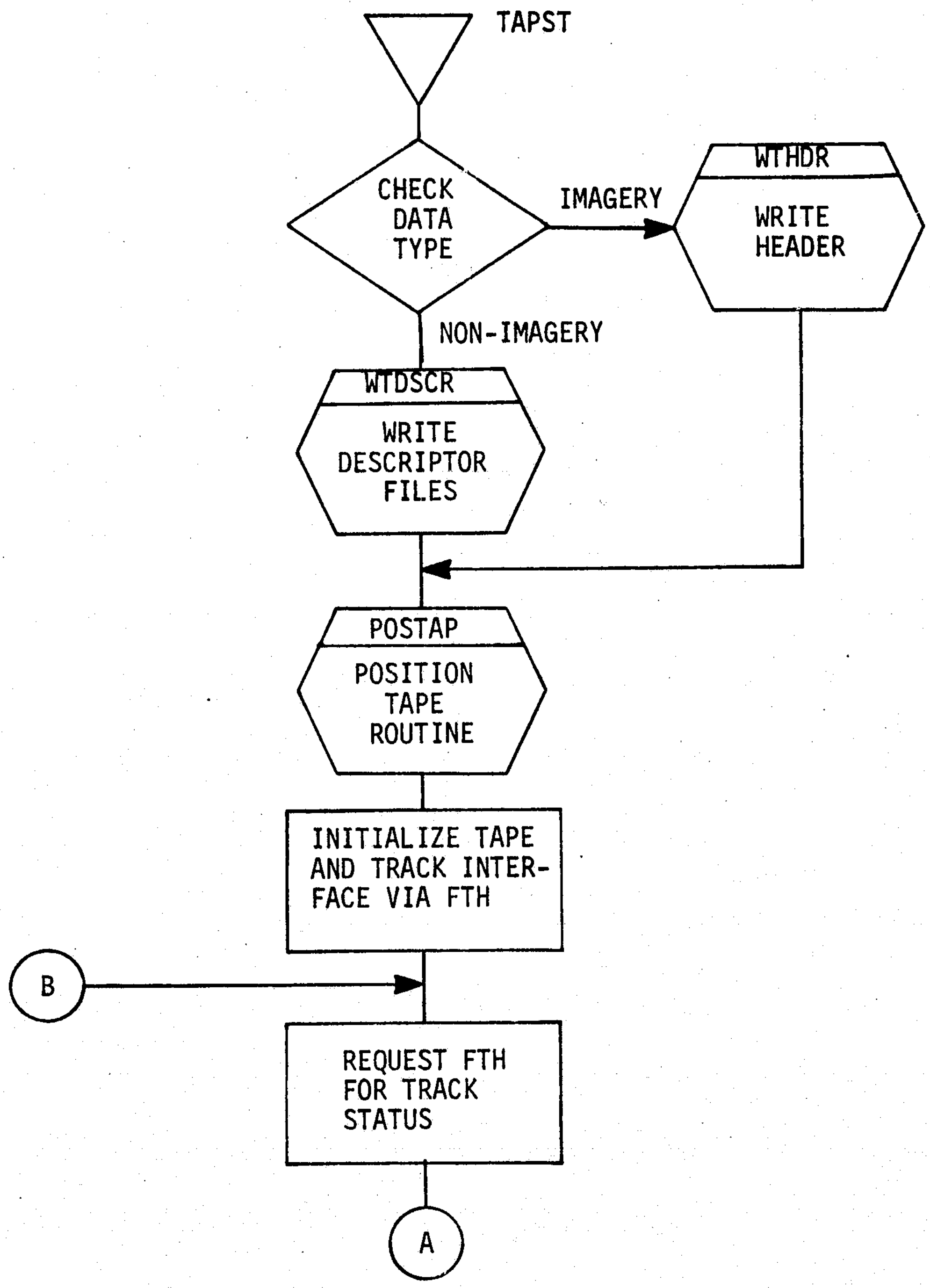


Figure 3-5 (24 of 30)

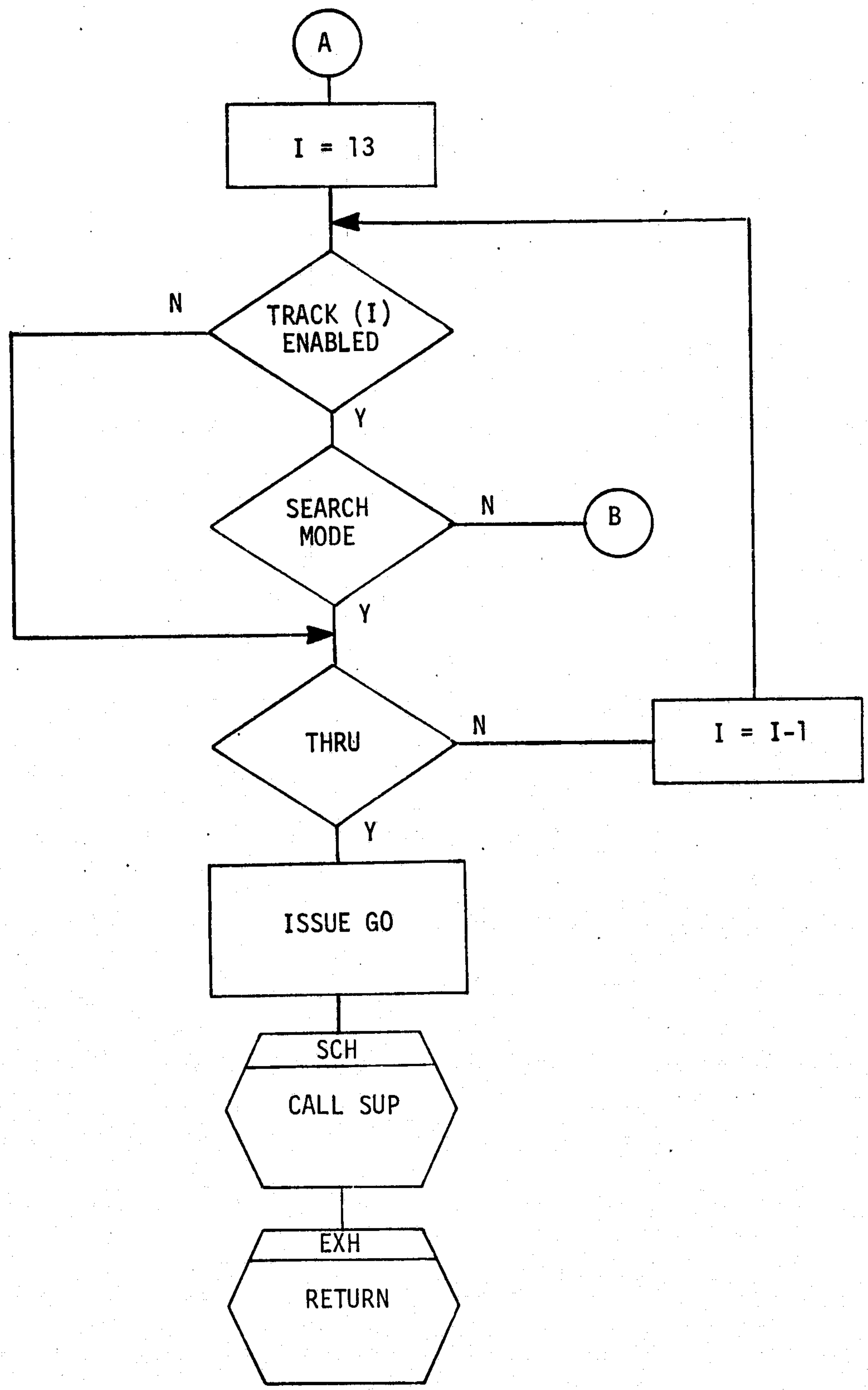


Figure 3-5 (25 of 30)

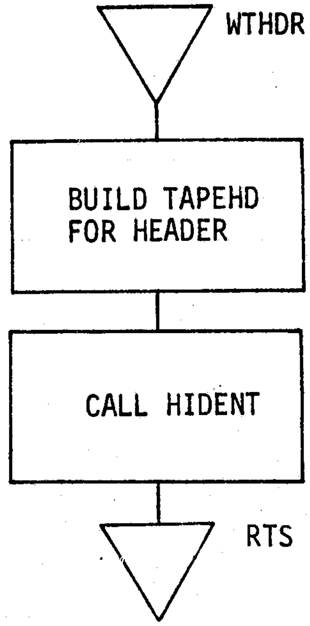


Figure 3-5 (26 of 30)

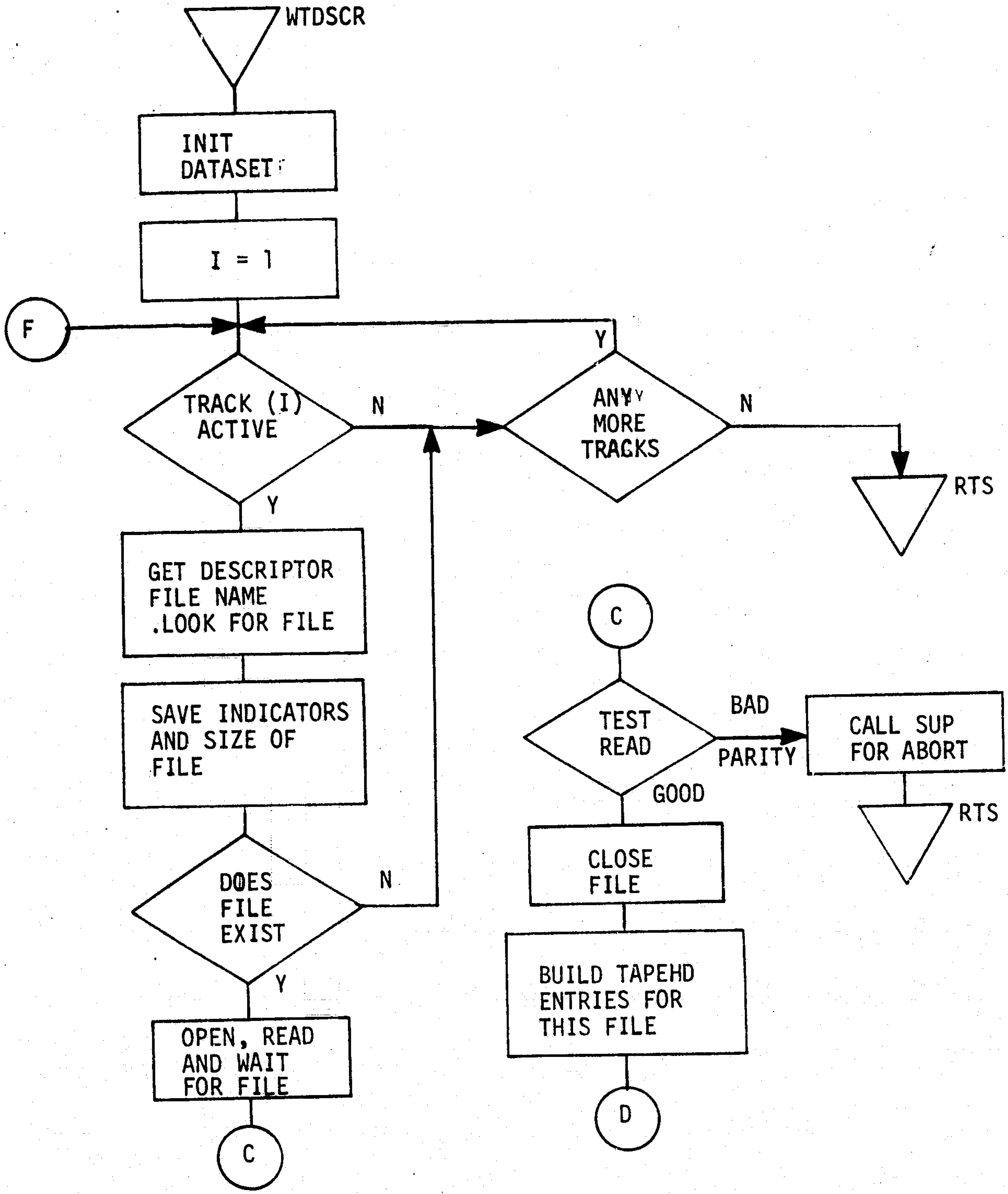


Figure 3-5 (27 of 30)

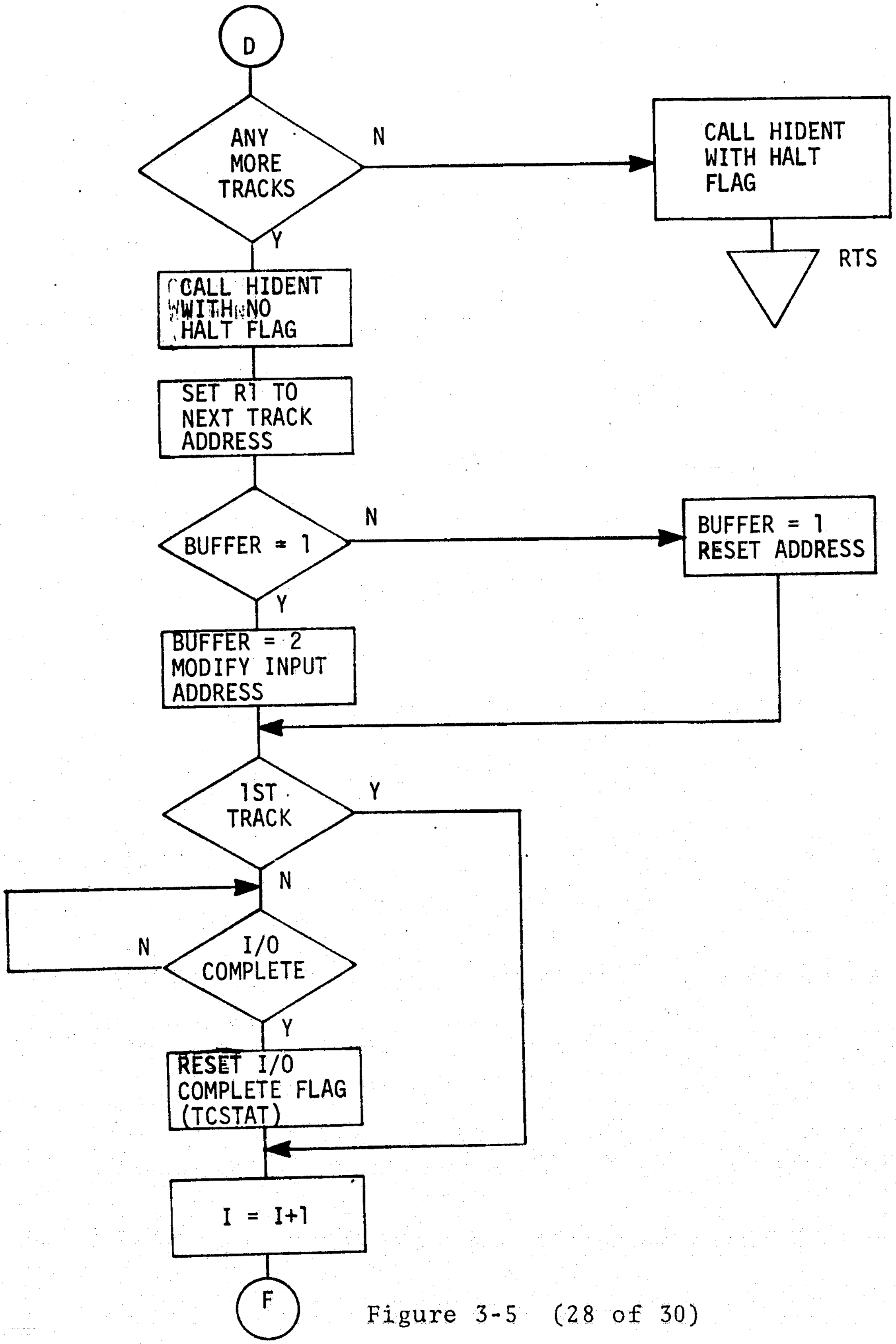


Figure 3-5 (28 of 30)

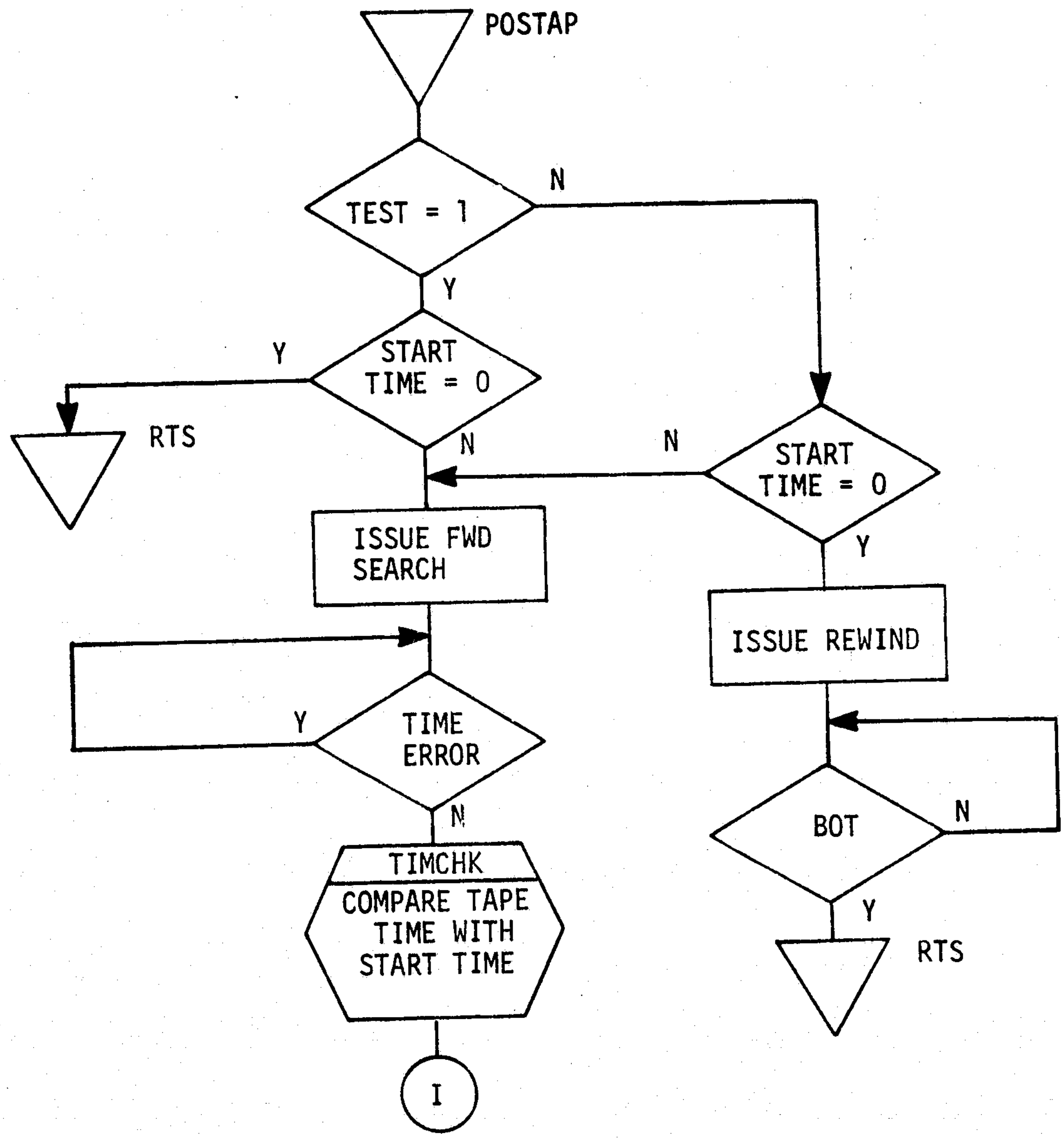


Figure 3-5 (29 of 30)

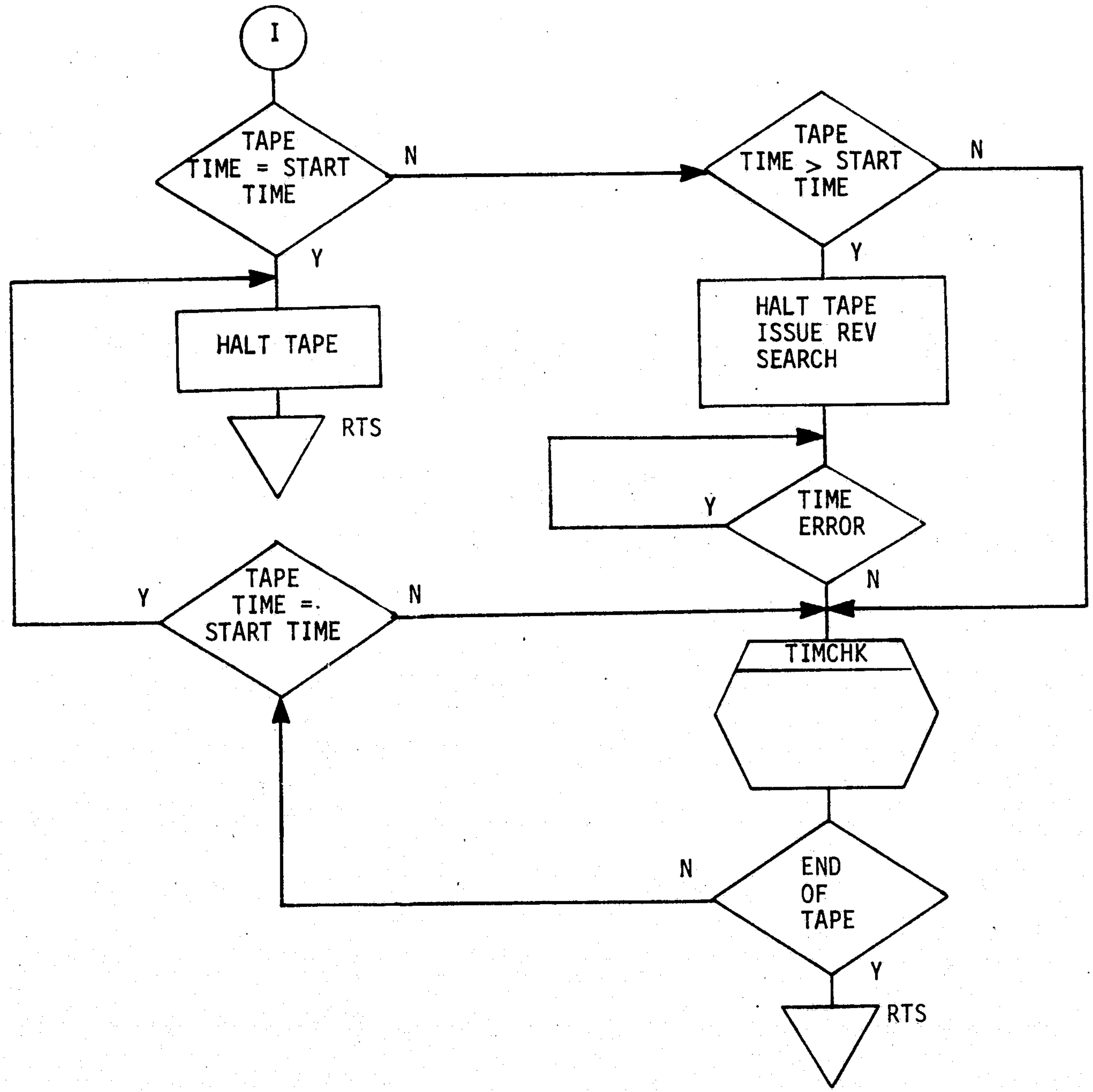


Figure 3-5 (30 of 30)

ADDRESS 764766

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED									LOCALE*	FILTER SELECT**					

*WHEN SET, FILTERS ARE SELECTED FROM FRONT PANEL
**FILTER CENTER FREQUENCY IS SET BY SELECTING ONE OF THE FOLLOWING BIT PATTERNS:

BITS 5 AND 4			BITS 3-0	BINARY	TAPE SPEED IN
0 & 1	1 & 0	1 & 1	3 2 1 0	MULTIPLIER	IPS = FILTER
78 Hz	-	-	1 1 1 1	1/128	-
156 Hz	15.6 Hz	-	1 1 1 0	1/64	-
312 Hz	31.2 Hz	-	1 1 0 1	1/32	1.875 = 55
625 Hz	62.5 Hz	-	1 1 0 0	1/16	3.75 = 54
1.25 kHz	125 Hz	12.5 Hz	1 0 1 1	1/8	7.5 = 53
2.5 kHz	250 Hz	25 Hz	1 0 1 0	1/4	15 = 52
5 kHz	500 kHz	50 Hz	1 0 0 1	1/2	30 = 51
10 kHz	1 kHz	100 Hz	1 0 0 0	1	60 = 50
20 kHz	2 kHz	200 Hz	0 1 1 1	2	120 = 47
40 kHz	4 kHz	400 Hz	0 1 1 0	4	200 = 46
80 kHz	8 kHz	800 Hz	0 1 0 1	8	-
0.16 MHz	16 kHz	1.6 kHz	0 1 0 0	16	-
0.32 MHz	32 kHz	3.2 kHz	0 0 1 1	32	-
0.64 MHz	64 kHz	6.4 kHz	0 0 1 0	64	-
1.28 MHz	128 kHz	12.8 kHz	0 0 0 1	128	-
2.56 MHz	256 kHz	25.6 kHz	0 0 0 0	256	-

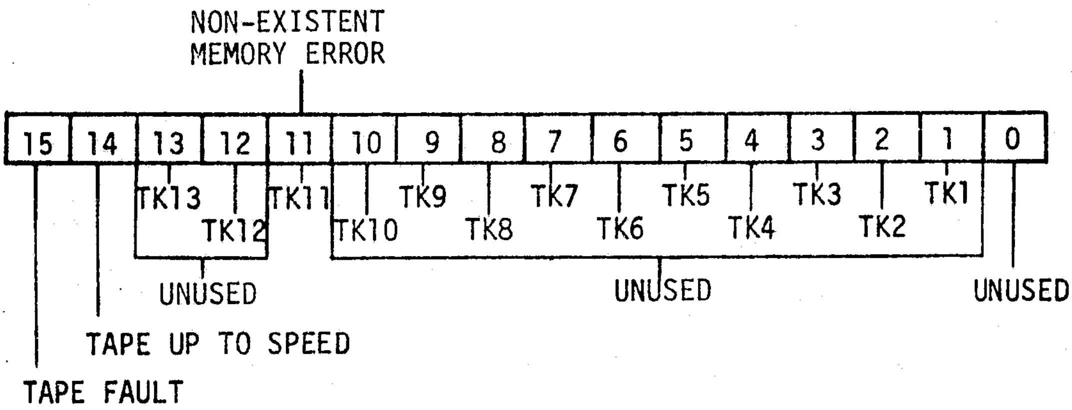
Figure 3-6 Time Code Translator Command Register

C. After positioning has been completed but before FTH is called to initialize the hardware track and tape interface, TAPST transfers the table IDPBA, built in TINIT, to the 525-line color display command and control registers.

3.2.3 <u>14-Track Interrupt Processor (FTIP)</u>. FTIP is the interrupt analysis routine for 14-track interrupts. It interfaces with the interrupt answering routines in FTH via a special scheduling scheme. After interrupt analysis for SEDS, vectored through location 174, FTIP calls MIPROG for further processing. FTIP picks up the interface control unit (ICU) status word and interrogate it for the possible error conditions, which are tape fault (bit 15) and nonexistent memory (bit 11). See figure 3-7. FTIP then checks the status of the track interface status registers (Nos. 1 and 2) to determine appropriate actions to be taken. This is done in the order shown below and illustrated in figure 3-8.

- Bit 13 - Buffer Full (Register 1)
- Bit 12 - Return to Lock (Register 1)
- Bit 11 - Overflow (Register 1)
- Bit 10 - Sync Error (Register 1)
- Bit 8 - Current Buffer Group (Register 1)
- Bits 10-11 - Interface Operating Mode (Register 2).

Actions taken from the various status conditions follow the same scheme used in the standard preprocessor.



ADDRESS = 764670

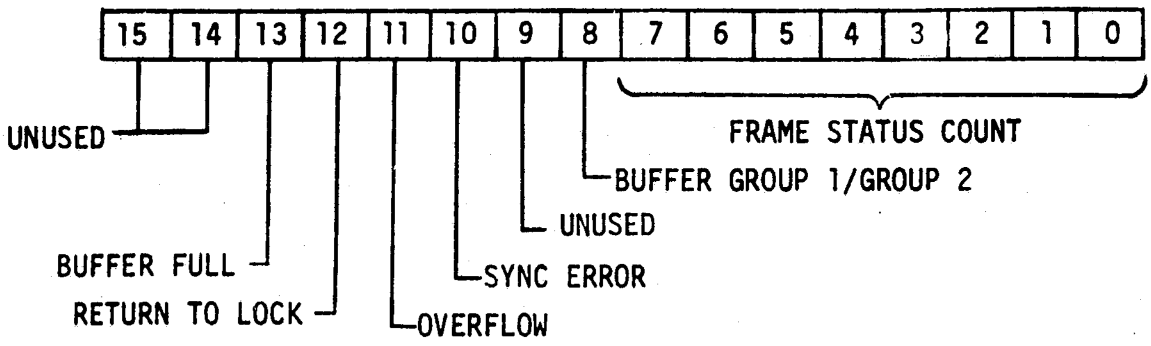
Bit 11 = The appropriate bit is set when the indicated track incurs a non-existent memory error. A non-existent memory error is detected when the data transfer circuitry waits more than 10 μsec for a slave response from the computer.

Bit 14 = The status bit is set when the capstan tachometer frequency is locked to the reference (internal or external) frequency. The bit is therefore set when the tape is up to speed.

Bit 15 = The bit is set when end-of-tape is reached, a broken tape occurs, or inoperative reel or capstan servo is detected. The bit is reset when the error condition is corrected.

Figure 3-7 Interface Control Unit Status Word

REGISTER NO. 1



Bits 7-0	Identify data frame currently being loaded; the count ranges from 0 to 254
Bit 8	Indicates which buffer group is currently being accessed; set indicates group 1
Bit 10	Indicates sync error condition when set
Bit 11	Indicates an overflow condition when set
Bit 12	Indicates an interrupt has been generated due to the interface returning to the rack mode when set
Bit 13	Indicates a buffer full condition when set

TRACK I/F NO.	ADDRESS
11	764500

Figure 3-8 Track Interface Register (1 of 2)

REGISTER NO. 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 11-10: INTERFACE OPERATING MODE

Bit 12: ENABLED/DISABLED

Bits 11-10 Indicate the present track interface operating state

Bit 11	Bit 10	
0	0	Halt mode
1	1	Lock Mode

Bit 12 Indicates condition of track interface. Set indicates enabled to perform transfers.

TRACK I/F NO.	ADDRESS
11	764530

Figure 3-8 (2 of 2)

3.3 IMAGERY PROCESSING (MIPROG)

Imagery processing is accomplished under the control of one module, the Main Imagery Processor (MIPROG). The capability of processing data input from the NOAA-supplied analog tapes was added to this module following its existing logic flow, and with only minor interleaving of new code. MIPROG is responsible for formatting the imagery data into the imagery universal format for output to 9-track CCT. In addition, it provides certain data quality oriented counters for VT05 display and summary logs.

3.3.1 Input Data. The processing of SEDS data begins when the buffer-full interrupt is detected and control is passed to MIPROG from FTIP. The data is pointed to the correct slot in the buffer area through hardware initialization. The rest of the format and data tag is inserted after the data input, to complete a data set. The data buffers are sized to accommodate one frame of each data type, infrared and visible. These two data types are placed in the buffer in a contiguous manner with no padding separating them (figure 3-9). The hardware interface always syncs on infrared and begins the data transfer to memory with it. Only one buffer-full interrupt is issued when both frames have been completely input. If there is a loss of sync for either of the data types, no buffer-full interrupts are issued until the proper sync is redetected. The analog interface generates, as the last byte of each data frame, a data tag. This tag identifies the data type that precedes it in the frame and flags, via error bit settings, data that has exceeded limits (see table 3-1).

3.3.2 MIPROG Processing. MIPROG handles the required data formatting and data quality checks for both the screening and edit passes. It is transparent to the 14-track interface, placing the raw data in memory regardless of which of the passes is being run. The data quality logic in MIPROG is used in the same manner for both runs, screening and edit. Two types of checks are performed on each buffer before it is output, and the data is output regardless of the results of the checks. The types of checks are described below.

A. Data Tag. The data tag is tested for each of the data types, infrared and visible, for each buffer-full

TABLE 3-1

DATA TAG VALUES

BIT	INDICATION
7	INFRARED DATA FRAME
6	VISIBLE DATA FRAME
5, 4	NOT USED
3	DC OFFSET REFERENCE CAL WAS GREATER THAN ALLOWED
2	DC OFFSET REFERENCE CAL WAS LESS THAN ALLOWED
1	GAIN OFFSET REFERENCE CAL WAS GREATER THAN ALLOWED
0	GAIN OFFSET REFERENCE CAL WAS LESS THAN ALLOWED

NO MORE THAN THREE BITS SHOULD BE SET IN ANY ONE TAG BYTE;
BITS 4 AND 5 MUST ALWAYS BE ZERO

processed. If the tag does not equal the expected values as follows, a corresponding counter is incremented:

- Infrared frame tag = bit 15 set
- Visable frame tag = bit 14 set.

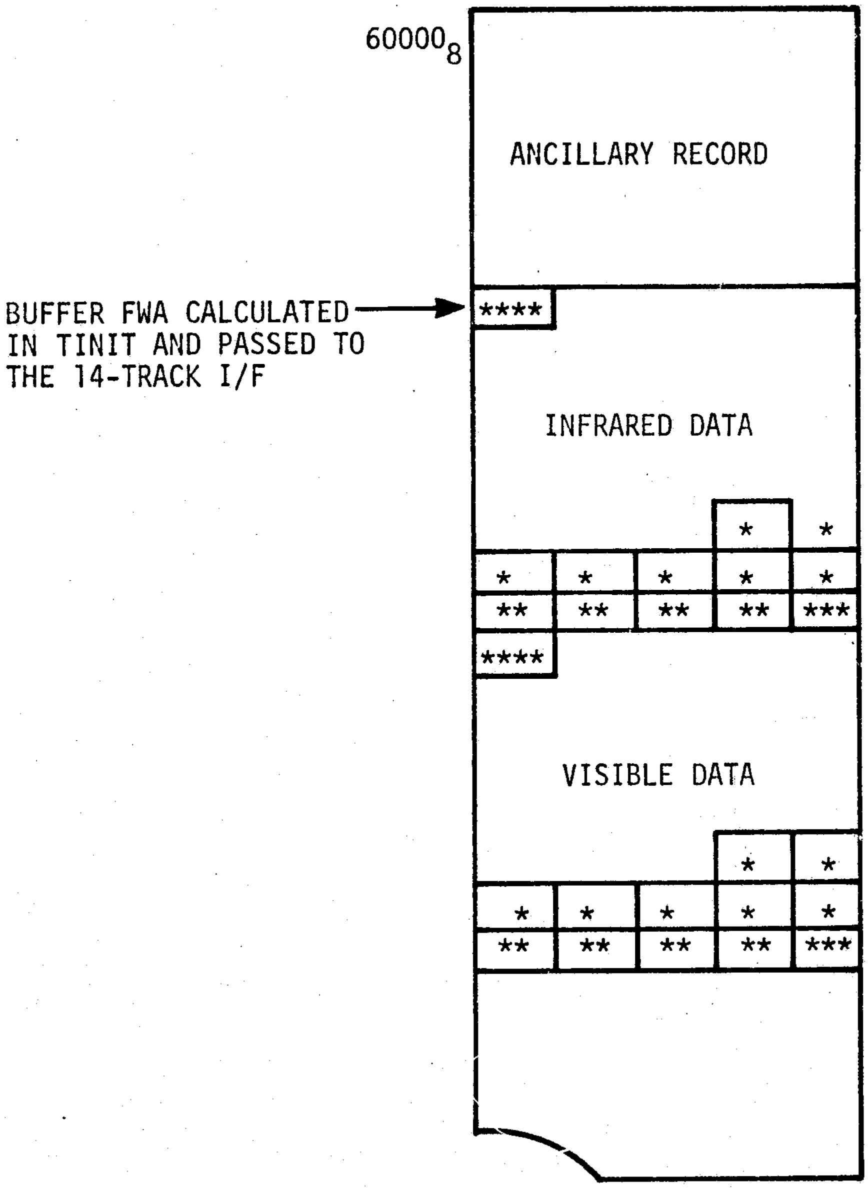
B. Time Delta. The time delta between the time the interrupt was issued for the current buffer and the previous buffer is calculated and compared to the nominal difference expected (150 ms). If the difference exceeds ±5 ms, the corresponding error counter is incremented.

The analog interface will take care of all the data checking and formatting described above. It will also generate the data tag to validate the data and flag what type of data it is, infrared or visible.

3.3.3 Processing of SEDS Data. MIPROG handles the necessary formatting for the edit run of the SEDS preprocessor. The matter of which pass -- edit, or screening, or both -- is being run is transparent to the 14-track interface. MIPROG will handle each run in the following manner.

A. Screening Run. For the SEDS screening run, the 525-line display is functioning in Monitor Mode, so that when a buffer-full interrupt is received from the 14-track, the data is already on the screen. Thus, the only remaining task is to do the data quality checking. The loss of data from loss of lock or sync is accomplished by comparing the time of the interrupt to that of the previous scan. The optimum delta is 150 ms and any time differences exceeding ±5 ms are flagged as errors on the VT05 data quality display. Following the above processing in MIPROG for the screening run, the program returns to the idle loop in IMAGI to await the next buffer-full interrupt.

B. Edit Run. The outputs received from the screening pass (see COUR in section 4) are used to initialize the preprocessor to edit the area desired from the 14-track. In MIPROG, for the edit run, when the buffer-full interrupt is



*SEVEN CALIBRATION VOLTAGE STEPS
**FOUR TEMPERATURE SENSOR SAMPLES
***DATA TAG
****SPACE SCAN SAMPLE

TOTAL SAMPLES (BYTES) INPUT FOR ANY PARTICULAR RUN IS USED TO INITIALIZE THE 14-TRACK, AND IS CALCULATED AS FOLLOWS:

SAMPLES/BUFFER = [(STOPPX-STRTPX)+14]2

Figure 3-9 Unprocessed Raw Data Buffer

issued, the buffer will be logged to 9-track CCT. The format of the tape will follow the current established criteria in the preprocessor (see paragraphs 4.4.1.6,E and F of *PHO-TR545, Preprocessor Subsystem Software Design Document*). The flow and logic for Category 1 processing that existed previously in MIPROG is followed with the exception of one step special cased for SEDS. The FWA of the area containing the second data type is reduced by two to include the last two bytes of the first data type. This word is the place where the record number is overlaid on the last two bytes of the first record after the first record has been logged (see DIRTY$).

C. Edit While Screening. During this processing phase, both the procedures for screening and editing are combined. The processing is controlled by the edit portion of the subsystem. Review the edit run section for further details.

3.4 PROCESSING SUPPORT SOFTWARE

The processing support software section of the subsystem provides program service for operations which are common to imagery, non-imagery, and SEDS processing. These support programs and their functions are as follows.

A. HIDENT. Writes data onto either 9-track CCT or high-density tape.

B. PTIME. Determines time.

C. SUP. Responds to control actions in the systems, and provides end-of-job cleanup.

D. ERTB. Collects status and error conditions from primary CPU during data transfer.

E. ERTH. Formats and outputs the data quality display once every 3 seconds.

These routines are mostly used in their previous configuration for implementation of the SEDS task. For a detailed description of these routines, see paragraph 4.5 of PHO-TR545.

3.4.1 High-Density Tape Processor (HIDENT). HIDENT provides the means for handling the orderly scheduling of output requests to 9-track tape for the SEDS data. The routine is also entered to analyze and take the appropriate action following any interrupt issued at the completion of an output function. This program will handle the receipt of single and multirequest data transfers and initiate 9-track output.

3.4.2 Time Program (PTIME). PTIME provides a means for converting IRIG-B time in 48-bit binary or formatted binary. This program is called by any user requiring the converted current IRIG-B time. No parameters are passed if the user wants current IRIG-B time. If the user wants some other IRIG-B time converted he should move the unformatted time in the first four words of the TIMTAB Table.

3.4.3 Support Control Program (SUP). This program provides the task control support functions. These functions include abort processing, end-of-job cleanup, enabling and disabling of tracks and channels, termination of the output tape, and job start lock check. SUP is given control with a parameter block containing the command code for processing. For enabling and disabling of tracks and channels, this block also contains track and channel numbers. These parameters are range-checked and, if they are out of range, DIGER is called to output an invalid input advisory on the VT05. Otherwise, the appropriate subroutine is called to process the code.

3.4.4 Error Recording Table Builder (ERTB). The primary purpose of ERTB is to collect status and error conditions occurring during data transfer. This data is used to generate the ERT Table, which is used for data quality displaying on the VT05 during processing.

3.4.5 Error Recording Table Output (ERTH). This routine, when processing is initiated, is scheduled to run at 1-second intervals. ERTH picks up the error tables built by ERTB; it logs them, and interfaces with the VT05 routines to build and output them as the data quality display.

3.5 PROCESSING UTILITY PROGRAMS

Several standalone programs in the Processing Software Subsystem provide data generation and examination facilities, as follows.

- Descriptor File Generator (GDESCR)
- Card Image File Generator for Disk or Tape (TMAKER)
- File Dump Utility Program (PRT).

These utilities remain available for possible future use in SEDS processing.

SECTION 4

OPERATOR INTERFACE SOFTWARE

4.1 GENERAL

The operator interface software is responsible for three basic functions:

- Providing an easy means for the operator to update and display parameters that affect the processing of the input data
- Providing a method for the operator to exercise control over the system by entering predefined commands into the system
- Displaying parameters pertinent to the quality of the data received, both in real-time on the CRT and post-run on the high-speed printer.

4.2 ROUTINES

To accomplish the above functions, eight existing routines have been defined. Several are limited to one function and several have overlapping functions. These routines and the functions they perform are as follows.

A. Parameter Input Routine (PARIN). Takes parameter inputs from cards or keyboard and displays them on the VT05.

B. Parameter Test Routine (PRMTST). Validates the input parameters and updates the tables used by PARIN. Runs as a subroutine to PARIN.

C. Command Processor Routine (COMP). Decodes and buffers all keyboard inputs. If the input is a parameter change, it schedules PARIN; if the input is a command, it calls the appropriate routine.

D. Display Generator Routine (DICER). Takes display commands from PARIN, PRMTST, COMP, and DQUP and generates a CRT image for the VT05.

E. Data Quality Data Base Update Routine (DQUP). Provides, on a real-time basis, the quality of the data being processed. The data base is then accessed by DIGER and SUMLOG for display and logging, respectively.

F. Summary Logger Routine (SUMLOG). Logs all initalization parameters and updates; operator actions (commands), and periods of data loss for later printouts. Also logs, on a periodic basis, capsule reports on the quality of the data received.

G. Courier Routine (COUR). Retrieves the files placed on the disk by SUMLOG and gives PTROT directives on the type of printout requested.

H. Summary Formatter Routine (PTROT). This routine is to the line printer what DIGER is to the VT05; formats and prints all summary reports.

Each of these routines is retained in its current format through the implementation of SEDS, with the exception of PARIN and COUR.

See the paragraph on task initialization (paragraph 3.2) for the modification to PARIN. COUR was expanded to output a single-page report following a screening run, which gives the delimiters for the area of interest as outlined by the operator on the 525-line color display (see figure 4-4 in paragraph 4.5). The delimiting values are picked up by COUR from a common data table, IAREA, built by the display interrupt handler INTER (see figure 4-3 in paragraph 4.5).

One new routine has also been added, for SEDS, to expand the operator interface with the preprocessor through the 525-line color display.

4.3 Job Parameter Input Routine (PARIN). See figure 4-1. The purpose of this routine is to give the operator a means of entering or changing parameters during the initialization phase of a preprocessing job, via card and VT05 inputs. For the SEDS task, a parameter was added to the normal preprocessor JOB card to indicate whether the run is to be a screening or edit pass. If this parameter is present (see Table 4-1), the routine expects the third card in the deck to be a SEDS parameter card (see table 4-2). The "US" card contains the following fields to aid initialization.

A. Day or Night Run. The imagery header is built in TINIT and will reflect whether it precedes data from a day or night pass.

B. Sample Window (Start and Stop Pixels). As input from the screening run, the sample window of data to be stripped from the 14-track tape is determined from the start and stop sample values for each data frame. At the end of a screening run, these values are calculated from the cursor positions which were input interactively during the first pass.

C. Display (Infrared or Visible). For the day pass tapes, the operator can specify which data is to be displayed, infrared or visible. The night tapes will contain only infrared data, so this parameter will not need to be set.

D. Sensor ID. This indicates which NOAA satellite the data was recorded from so that the proper ID can be placed in the header of the edit tape.

These parameters, along with internally defined constants, make up the variables and control functions needed to properly interface with the 14-track.

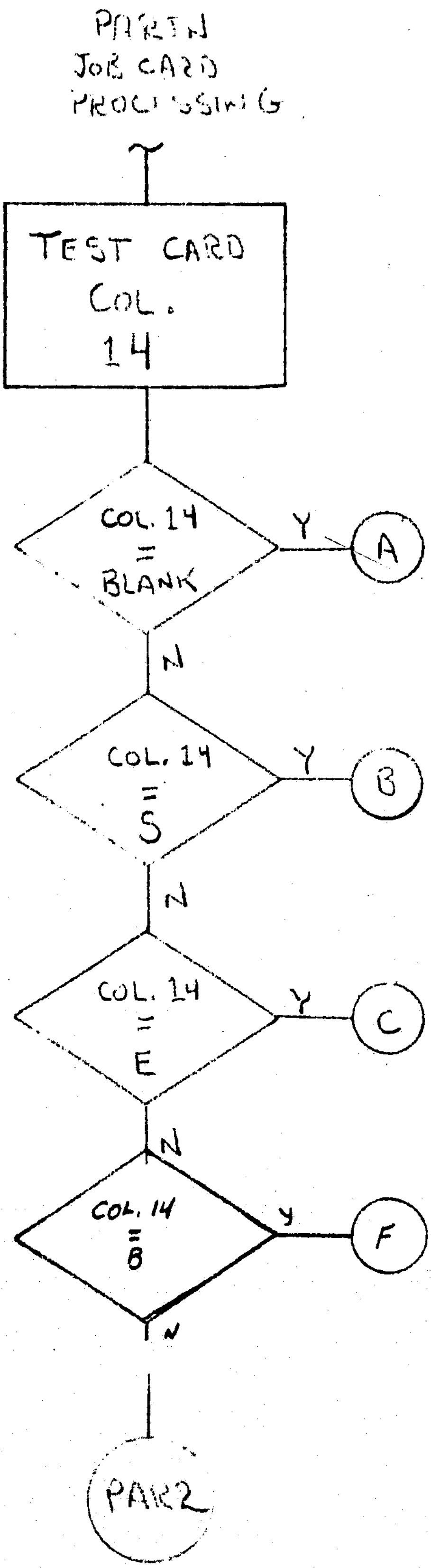


Figure 4-1 PARIN Flow (1 of 13)

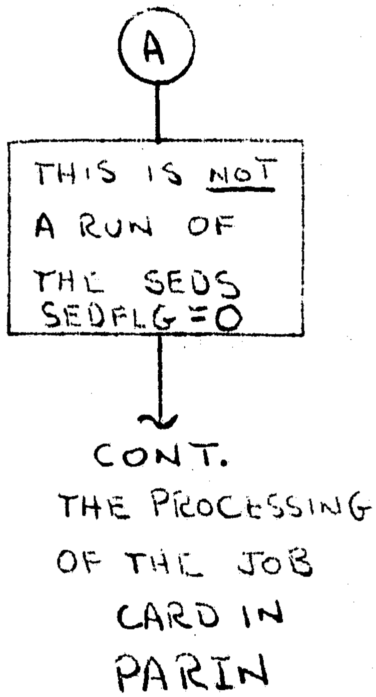


Figure 4-1 (2 of 13)

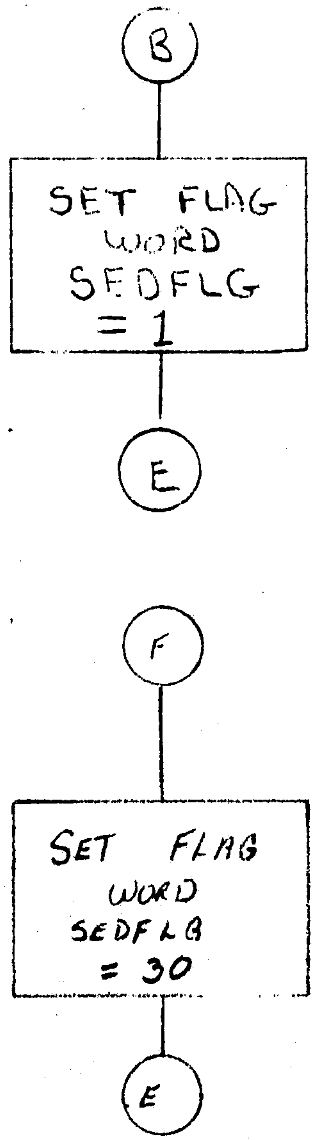
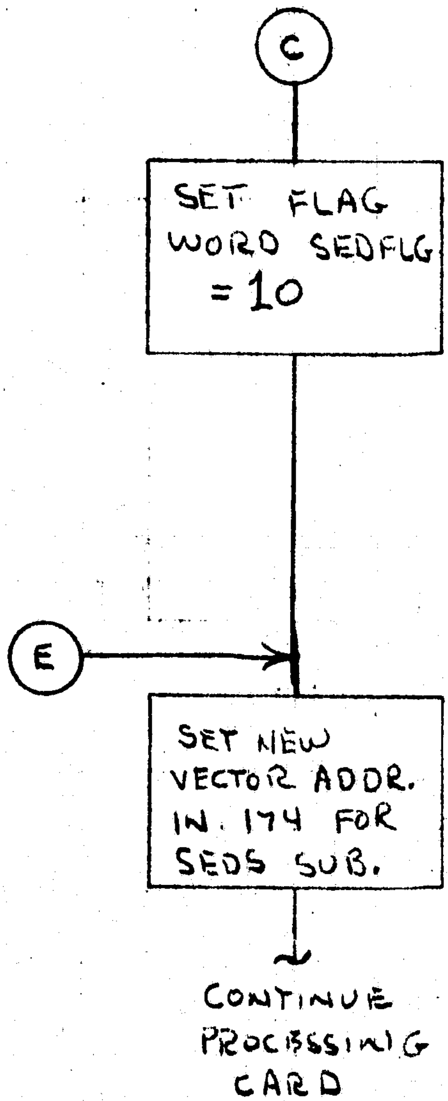


Figure 4-1 (3 of 13)



ORIGINAL PAGE IS OF POOR QUALITY

Figure 4-1 (4 of 13)

C 2

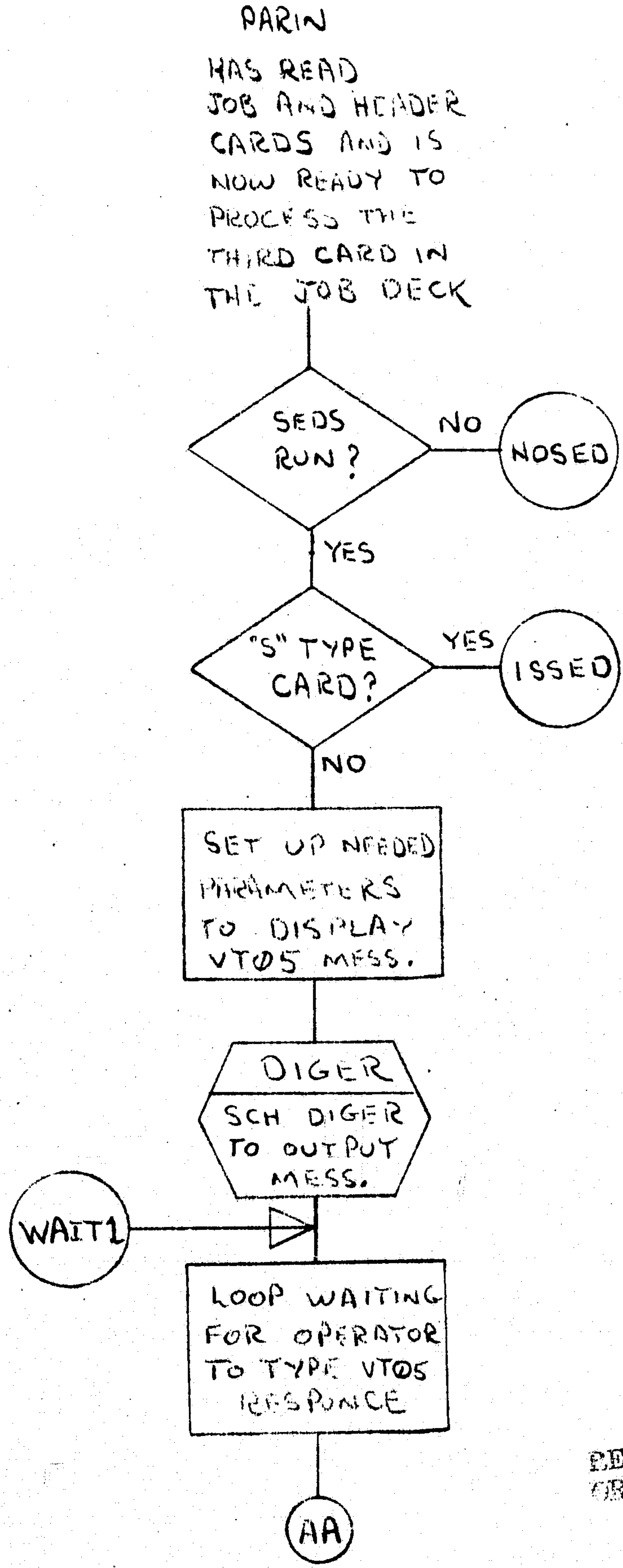


Figure 4-1 (5 of 13)

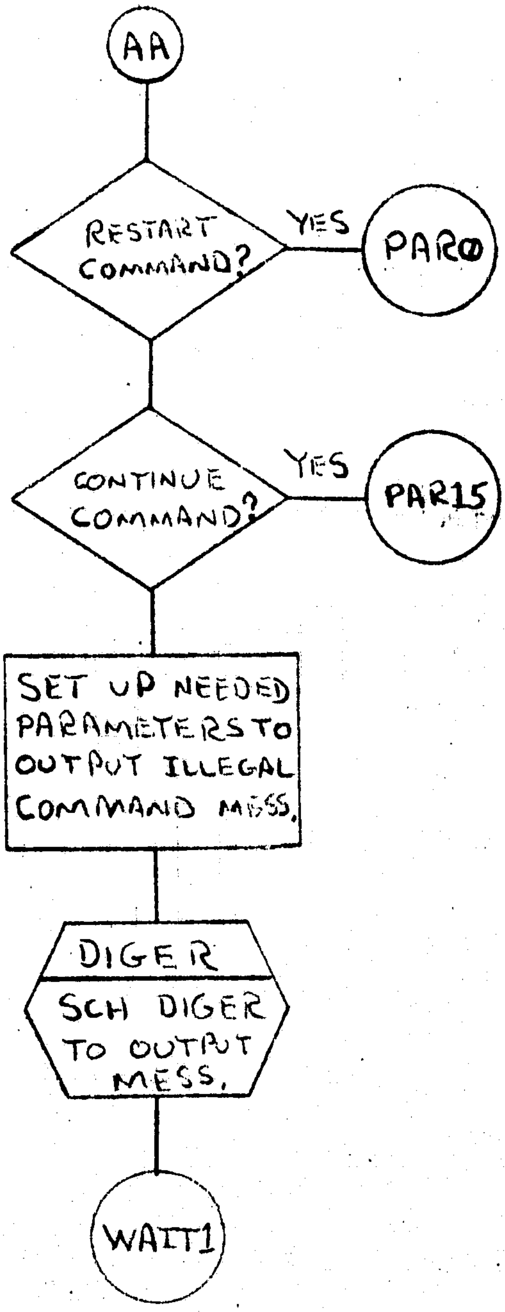


Figure 4-1 (6 of 13)

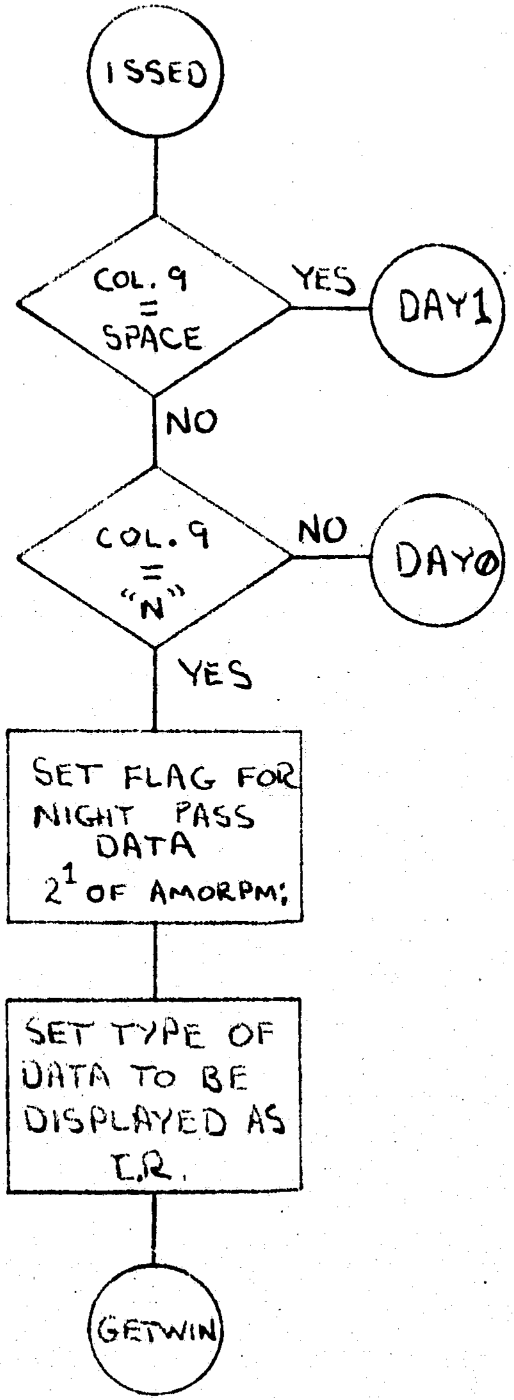


Figure 4-1 (7 of 13)

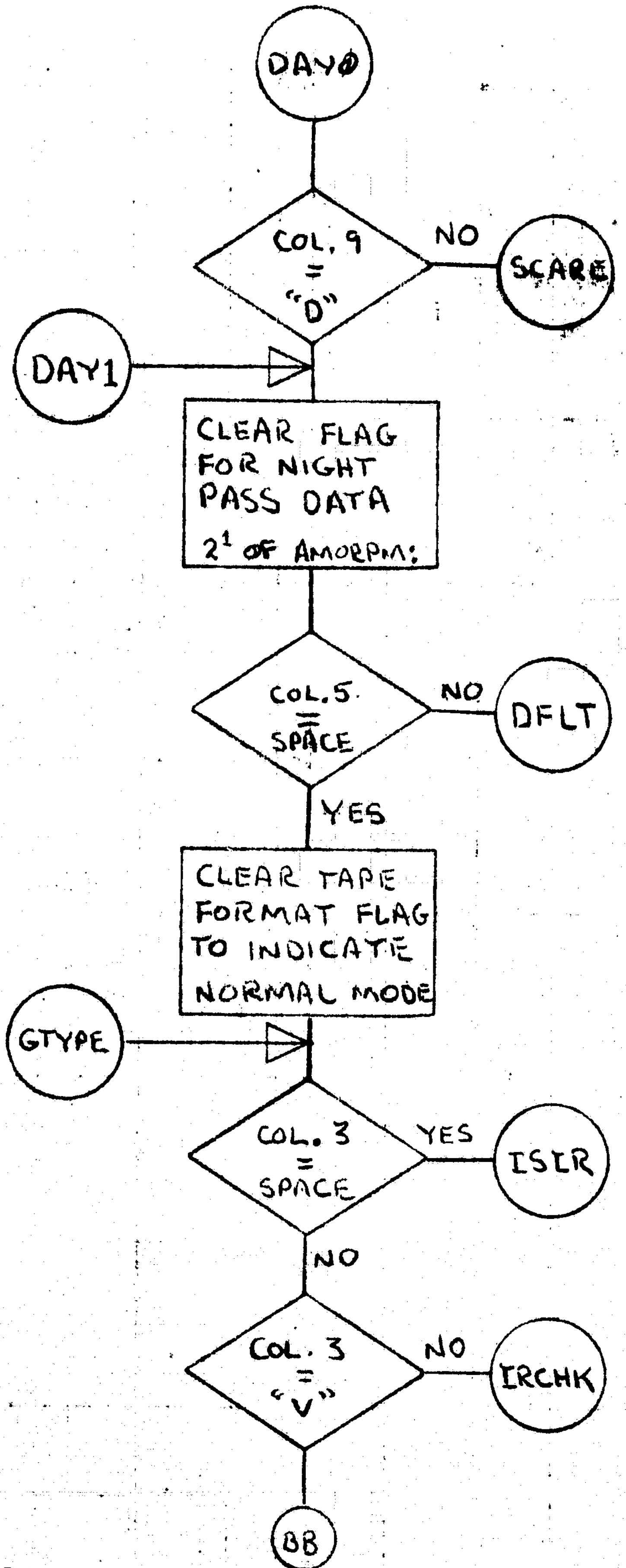


Figure 4-1 (8 of 13)

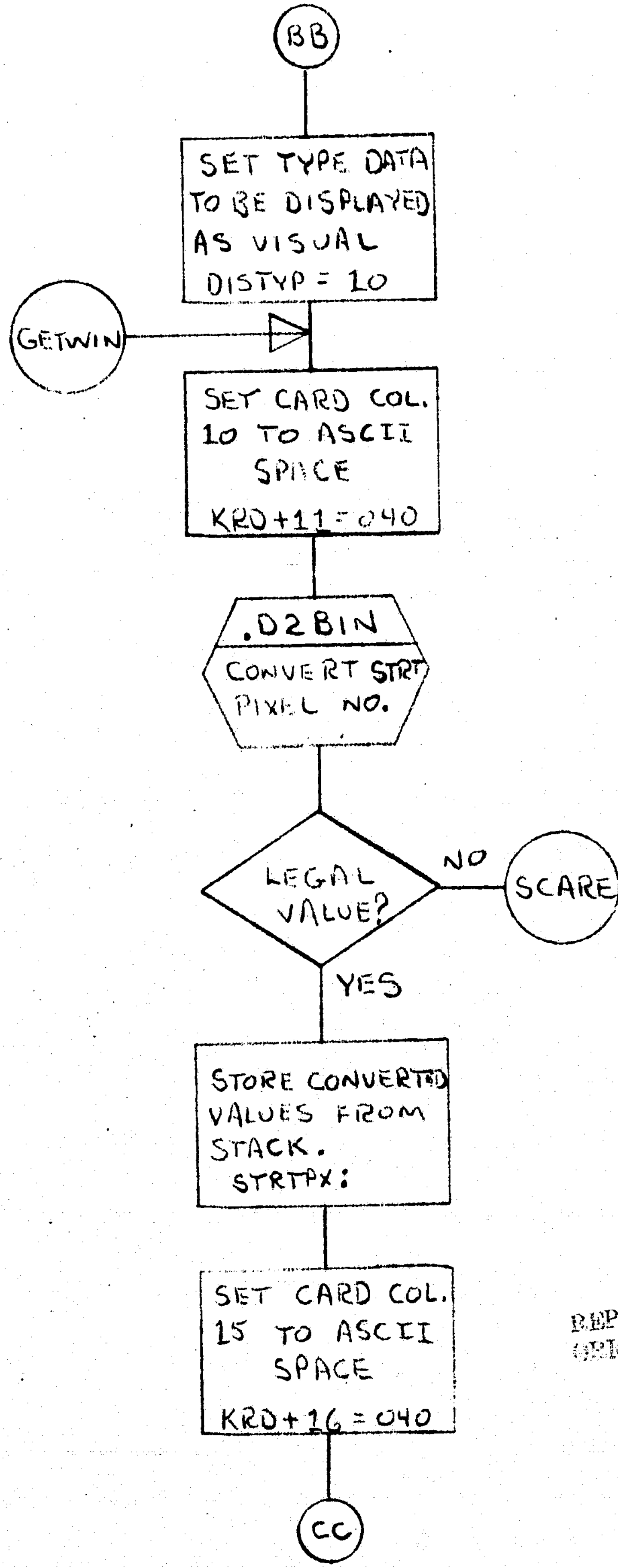
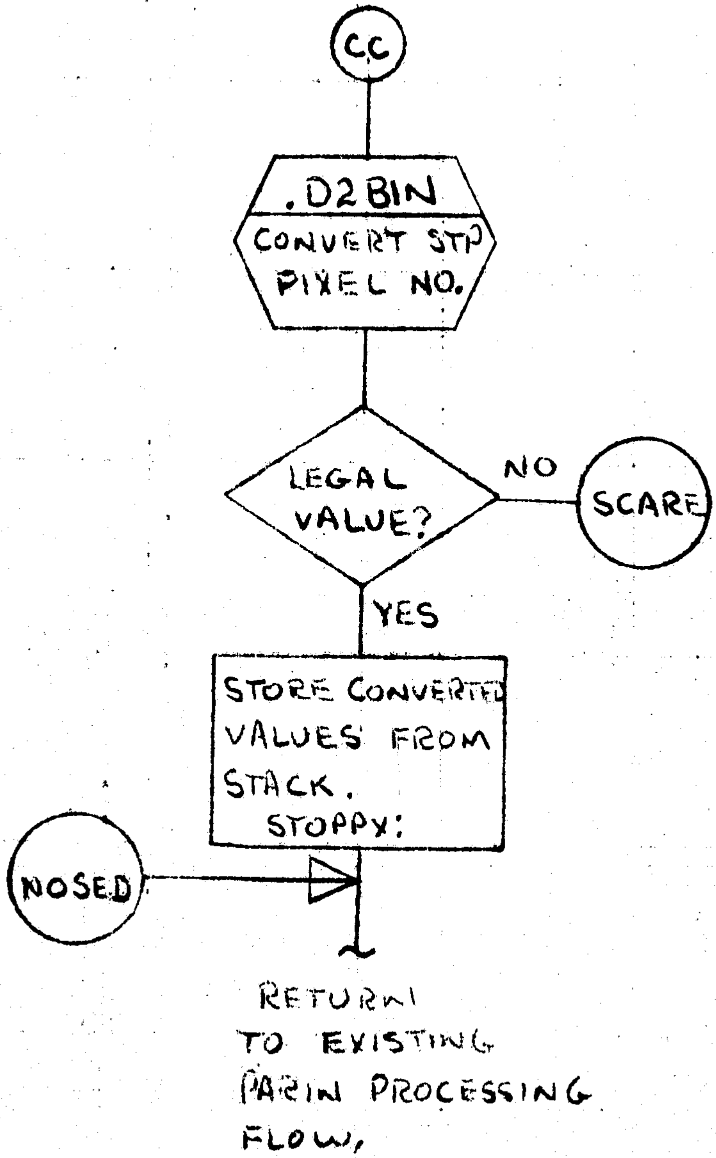


Figure 4-1 (9 of 13)



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 4-1 (10 of 13)

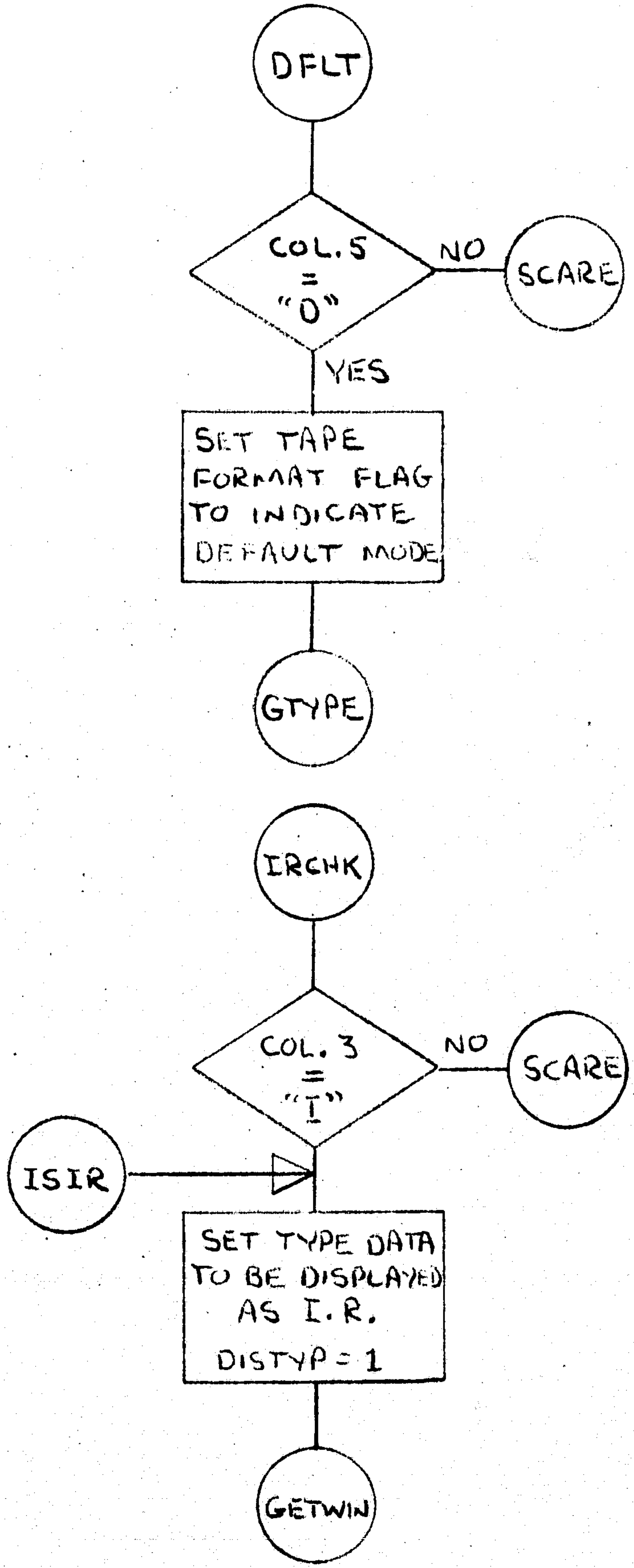


Figure 4-1 (11 of 13)

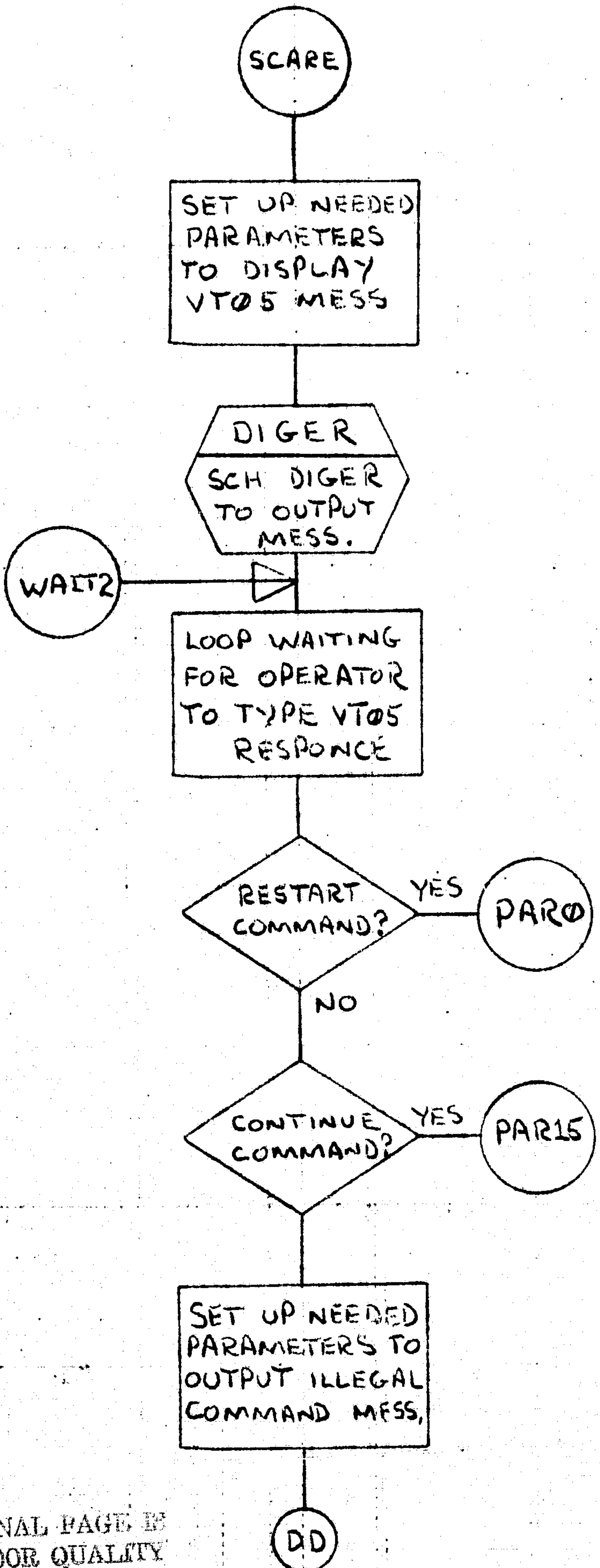


Figure 4-1 (12 of 13)

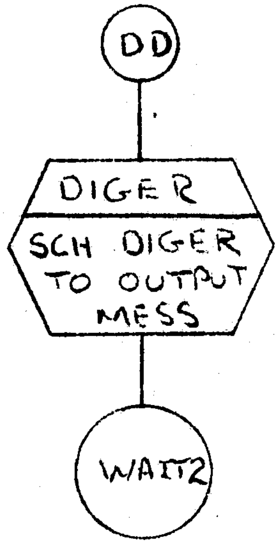


Figure 4-1 (13 of 13)

TABLE 4-1

SEDS PREPROCESSOR JOB CARD

COLUMN	ENTRY (NO DEFAULT)	RANGE	REMARKS
1	J	-	ERROR MSG DISPLAYED IF PARAM NOT PRESENT
3-12	FILE NAME AND EXTENSION	10 ASCII ALPHA-NUMERIC CHARS	PARAM MUST BE PRESENT; NAME IS UP TO 6 CHARS AND A PERIOD FOLLOWED BY 3-CHAR EXTENSION. CHARS USED INCLUDE BLANKS, ' E.G., S192.LAB
14	S-E-B*		
16	NO. OF TASKS	1-5	PARAM MUST BE PRESENT AND WITHIN RANGE
24	I OR N	-	PARAM MUST BE PRESENT; INDICATES TYPE OF DATA FOR THIS JOB (I = IMAGERY, N = NON-IMAGERY)
26	H OR 9	-	PARAM MUST BE PRESENT; INDICATES DATA OUTPUT DEVICE (H = HIGH DENSITY, 9 = 9-TRACK)
28	ASTERISK OR BLANK	-	* = NEW HD TAPE (USED ONLY WITH H ABOVE); BLANK = HD TAPE CONTAINING DATA
41-60	OUTPUT TAPE	1-20 ASCII ALPHA-NUMERIC CHARS	PARAM MUST BE PRESENT AND WITHIN RANGE

*S = SCREENING PASS, E = EDIT PASS, OR B = BOTH

TABLE 4-2

FORMAT OF S-CARD REQUIRED FOR RUN ON SEDS DATA

COL	ENTRY	DEFAULT	RANGE	REMARKS
1	S	--		S DENOTES A SEDS DATA CARD; REQUIRED ONLY WHEN COL 14 OF JOB CARD IS SET
3	IR OR VISIBLE	IR	I OR V	DENOTES DATA TO BE DISPLAYED ON SCREENING PASS (OR TYPE OF DATA ON TRACK IF 14-TRACK IN DEFAULT FORMAT)
5	14-TRACK FORMAT FLAG	NORMAL FORMAT	D	PRESENT ONLY IF IR AND VISIBLE DATA RESIDE ON SEPARATE TRACKS
7	DAY OR NIGHT PASS	D	D OR N	DENOTES IF DATA IS FROM A DAY OR NIGHT PASS
13-16	START PIXEL	1	1-3500	SPECIFIES START PIXEL NO. FOR DESIRED SAMPLE WINDOW
18-21	STOP PIXEL	3500	1-3500	SPECIFIES STOP PIXEL NO. FOR DESIRED SAMPLE WINDOW
	SENSOR ID	NOAA2	1-2	WITH SENSOR ID, SPECIFIES TO INPUT TO HEADER (NOAA3-1 OR NOAA3-2)
9	SENSOR ID	NOAA4	2-3	
11	SENSOR EXP	1	1-2	

4.4 Parameter Test Routine (PRMTST). See figure 4-2. The primary function of PRMTST, a subroutine called by PARIN, is to make appropriate validity checks on the job and task parameters input by the operator, either from cards or the VT05. For SEDS preprocessing, PRMTST will validate the following.

A. Pass Update. Checks to see whether an N or D has been input as the pass parameter. (N = night pass; D = day pass). If the correct parameter has not been input, an error message will appear on the VT05 to reflect an error condition in the pass field.

B. Vehicle Update. Determines the NOAA format to be processed.

C. Start Pixel. Confirms that start pixel number is greater than zero and less than 3490 (less than the stop pixel).

D. Stop Pixel. Confirms that stop pixel number is greater than zero, greater than the start pixel, and less than 3490.

E. Run Update. Verifies that the appropriate S, E or B parameter has been input as the type of review to be processed (S = screening/ E = edit; B = both).

If there exists an error condition in either of the fields checked by PRMTST, an error flag (the letter E) will be displayed on the VT05 adjacent to the field that is in error. The task will be halted until the correct parameter has been input.

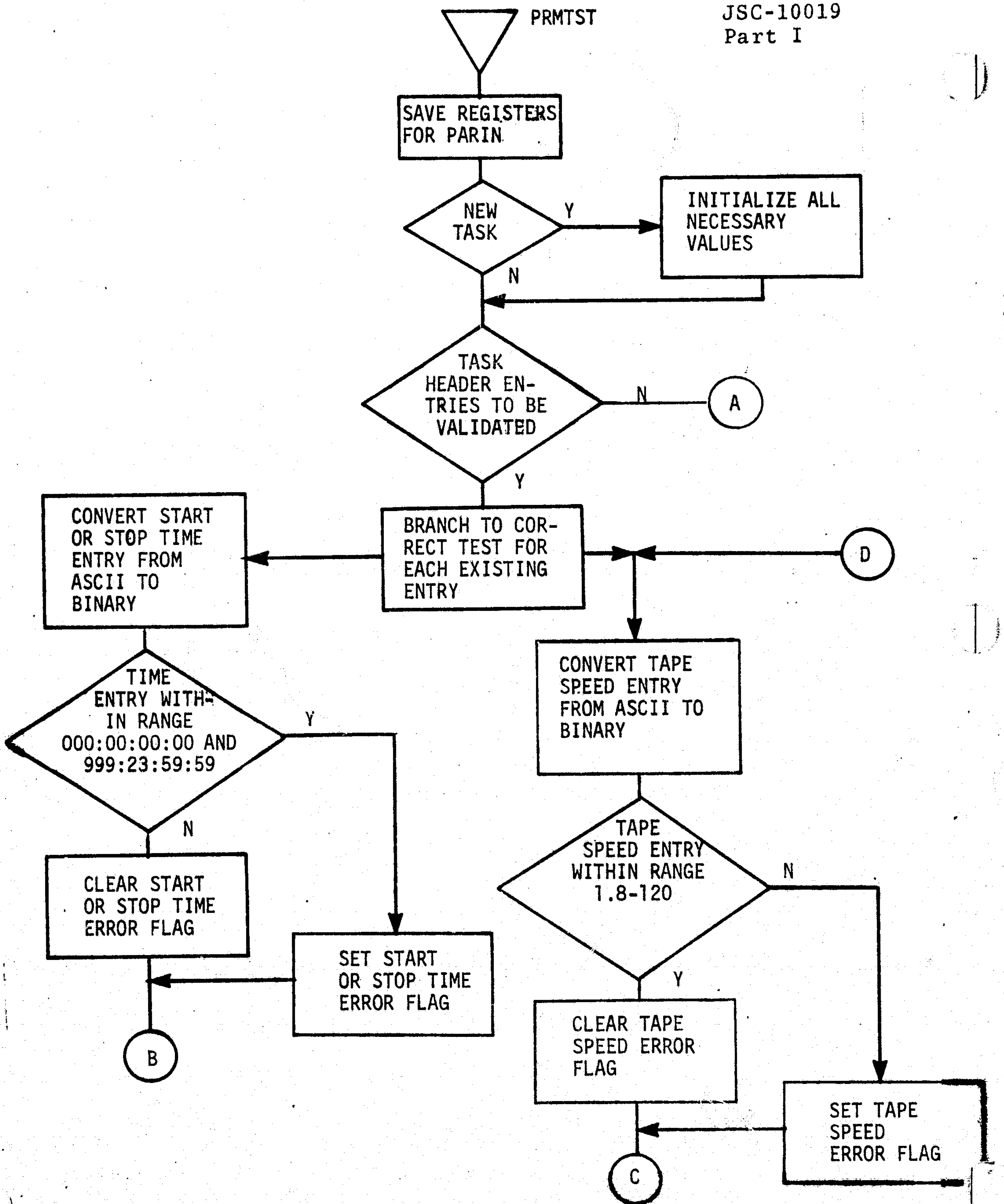


Figure 4-2 PRMTST Flow (1 of 4)

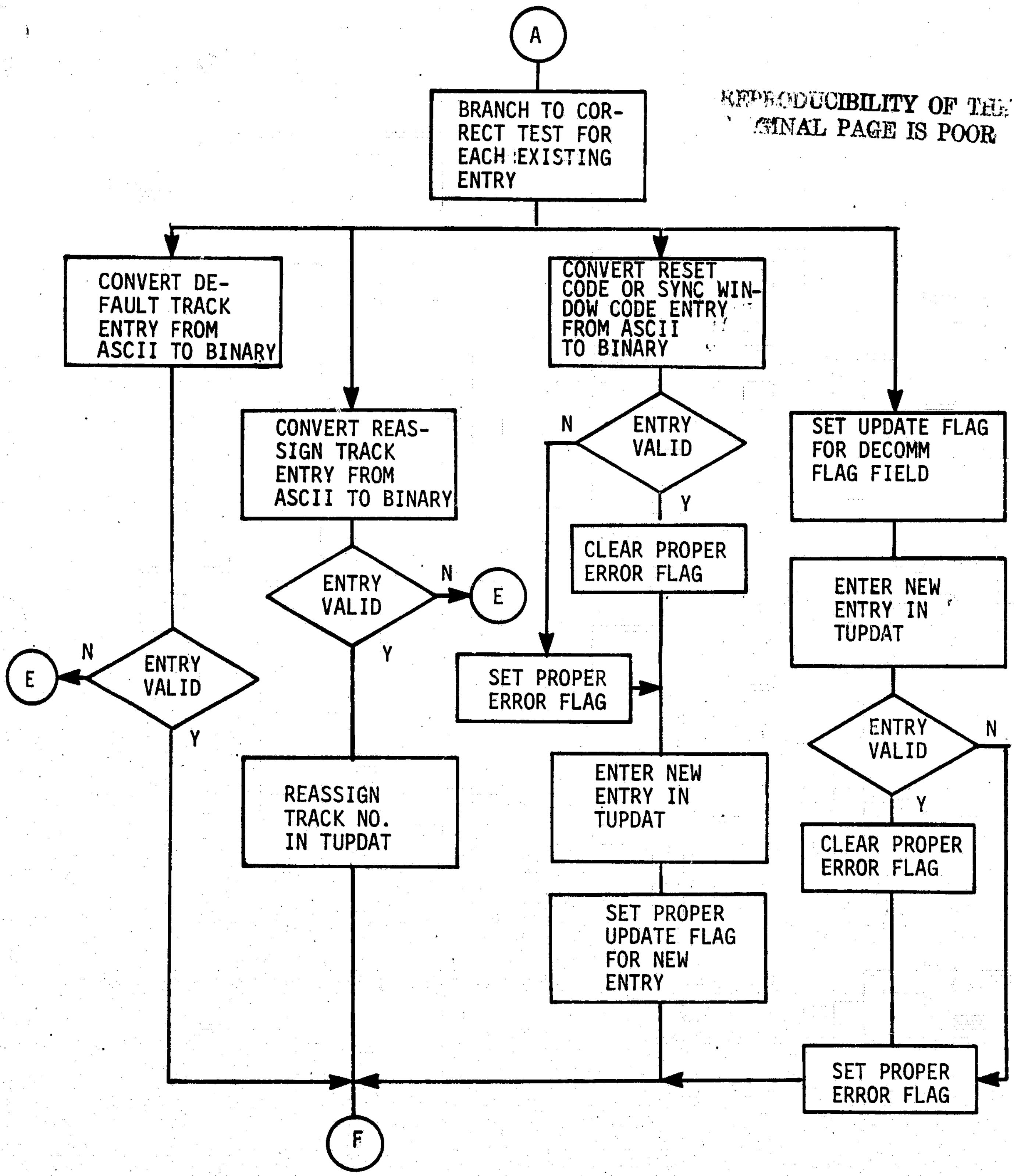


Figure 4-2 (2 of 4)

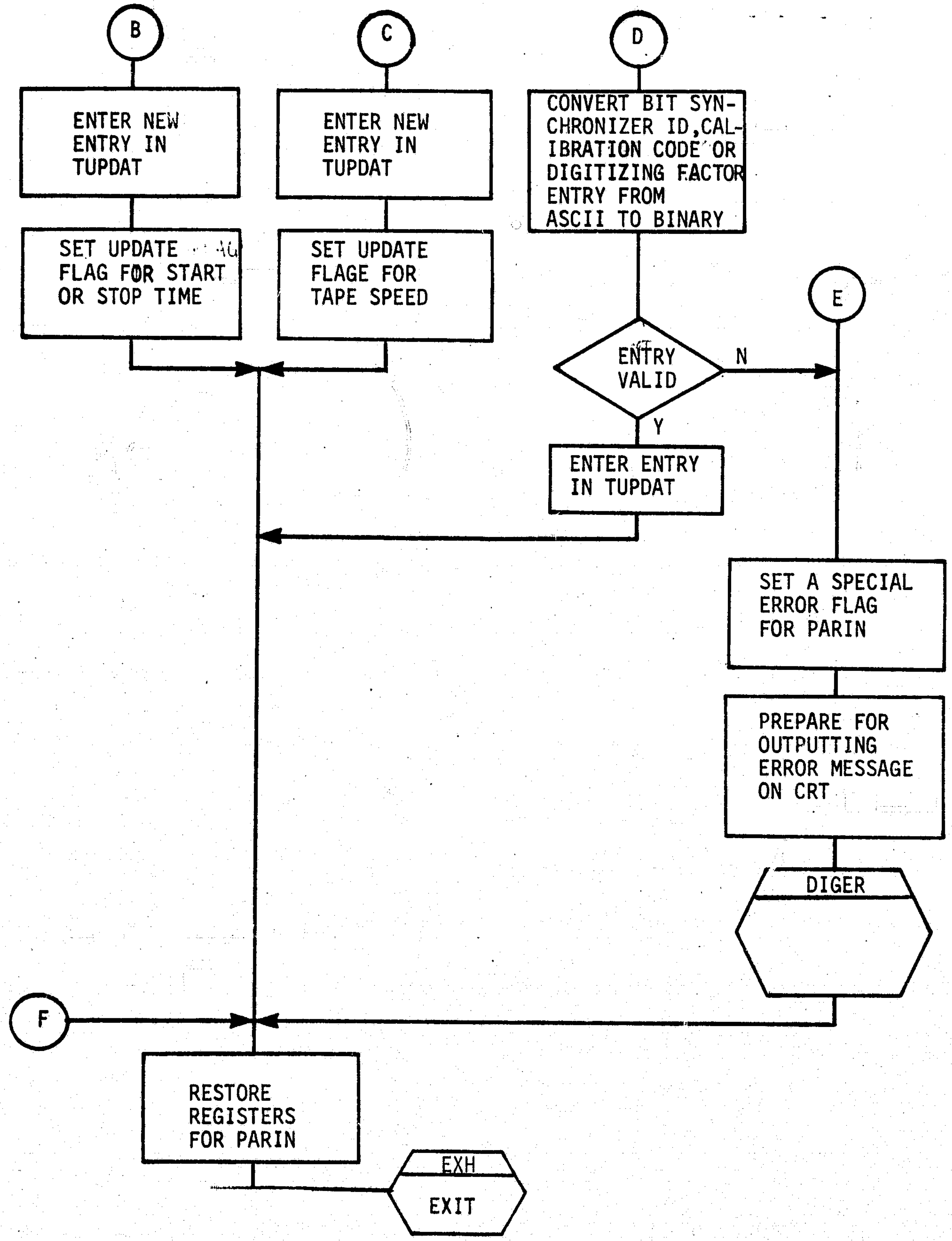


Figure 4-2 (3 of 4)

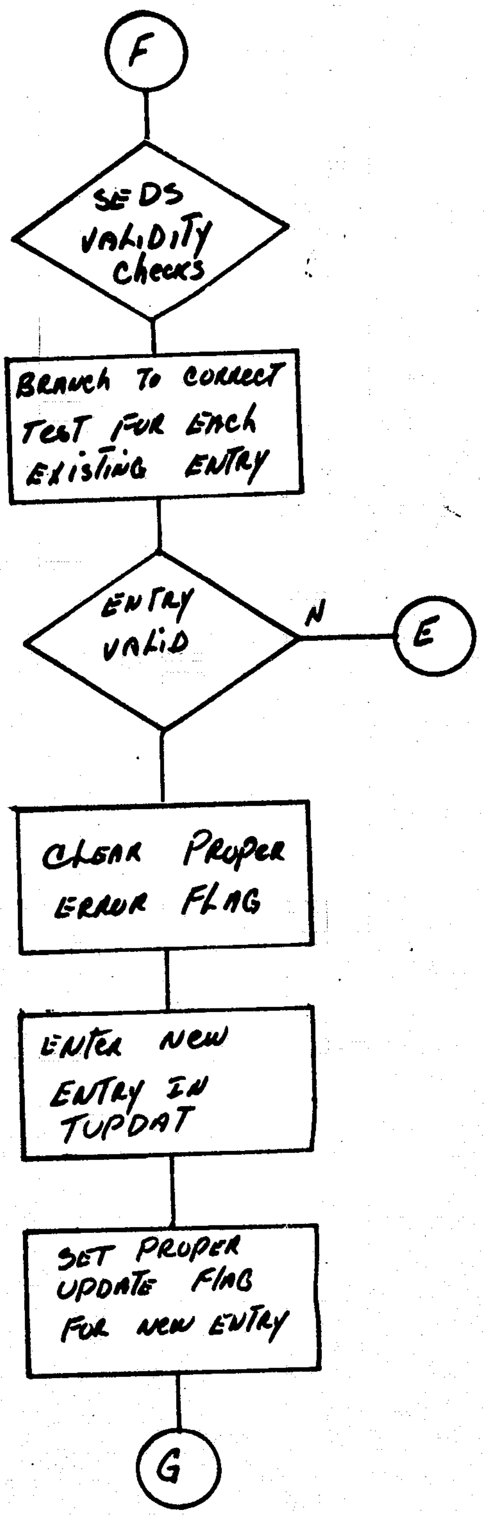


Figure 4-2 (4 of 4)

4.5 525-LINE COLOR DISPLAY INTERRUPT HANDLER ROUTINE (INTER)

See figures 4-3 and 4-4 and Table 4-3. This module was written for and exists only in the SEDS preprocessor. It handles all interrupts issued from the 525-line display and can be entered only in this manner. The routine examines all interrupts received during the screening run and takes the appropriate action. Any interrupts received during the edit run are ignored.

The interrupt vector for the 525-line color display is set in IMAGI, and the hardware command and format registers are initialized in TAPST (see paragraph 3.2.2). For the screening run, the display functions in Monitor Mode, which requires no transfer from memory to the display. In this mode, the display hardware monitors the UNIBUS until it detects the first word address that is initialized in its register, and then begins trapping data into its memory until the desired message length is satisfied. Using this method, the display functions on its own once it is initialized, requiring no additional interfacing, until the operator interrupt is received to halt the 14-track tape. Upon receiving an interrupt from the display, INTER processes it in the following basic steps.

A. INTER determines if the interrupt is legal, discarding any command complete interrupts during screening and all interrupts during editing. The first legal interrupt that can be processed is an operator-initiated interrupt to halt the 14-track tape. Legal operator interrupt request codes that initiate a response are the following thumb-wheel values.

- 00 - Halt 14-track
- 10 - Cursor defines upper boundary
- 11 - Cursor defines lower boundary
- 20 - Cursor defines right broundary
- 21 - Cursor defines left boundary

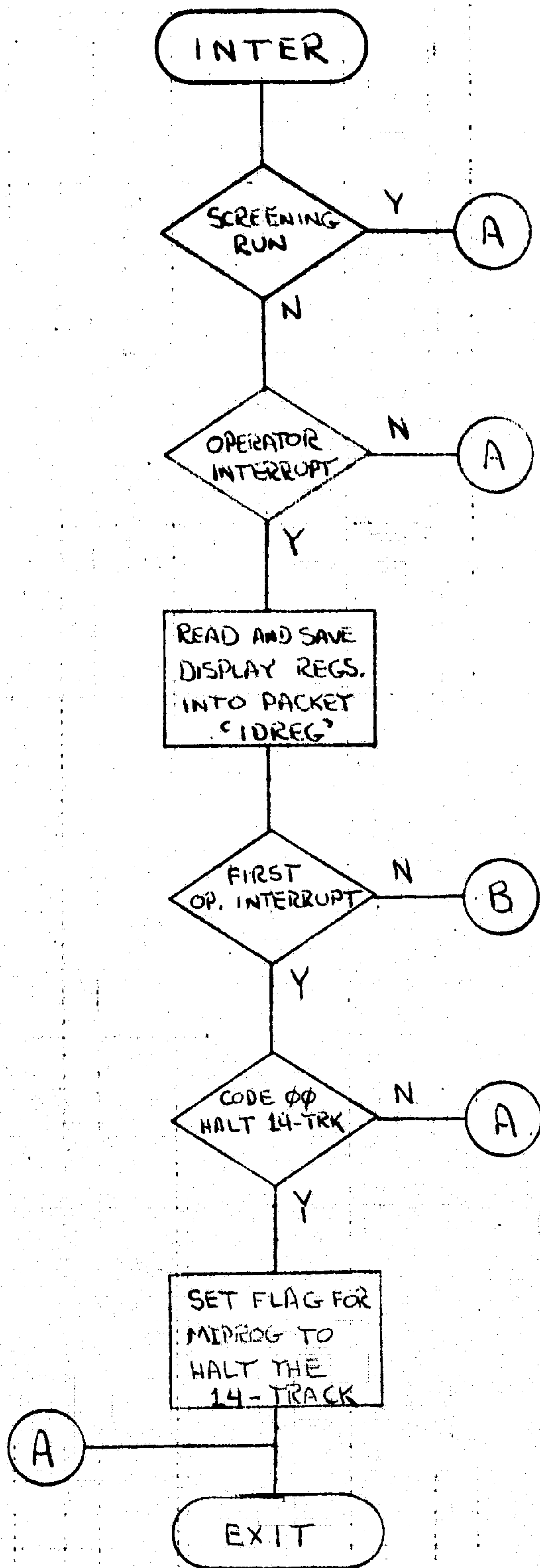


Figure 4-3 INTER Flow (1 of 10)

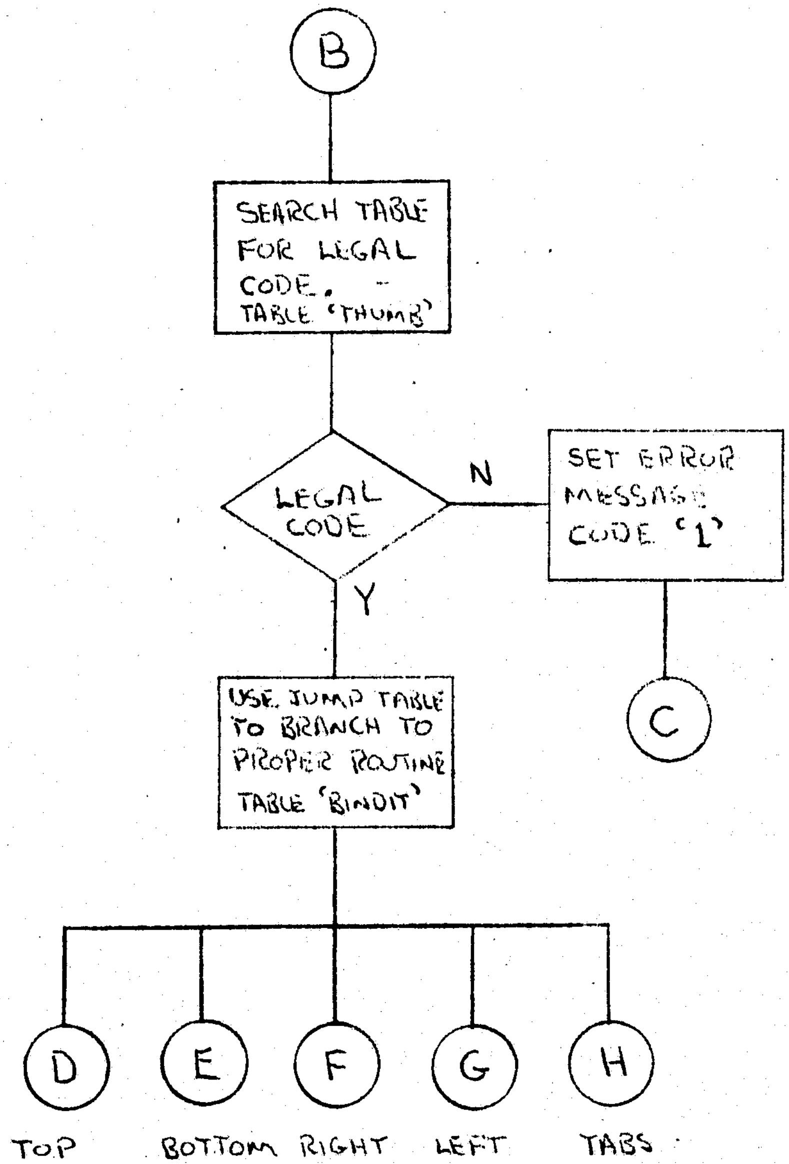


Figure 4-3 (2 of 10)

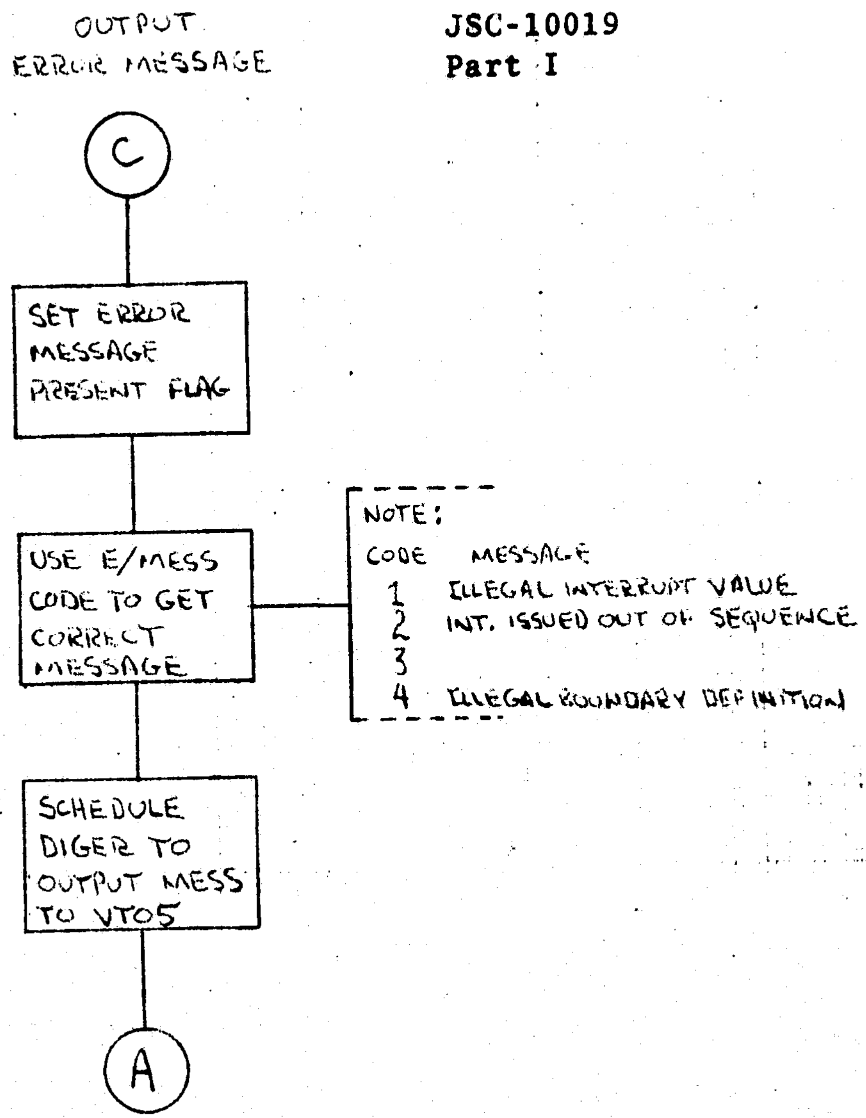


Figure 4-3 (3 of 10)

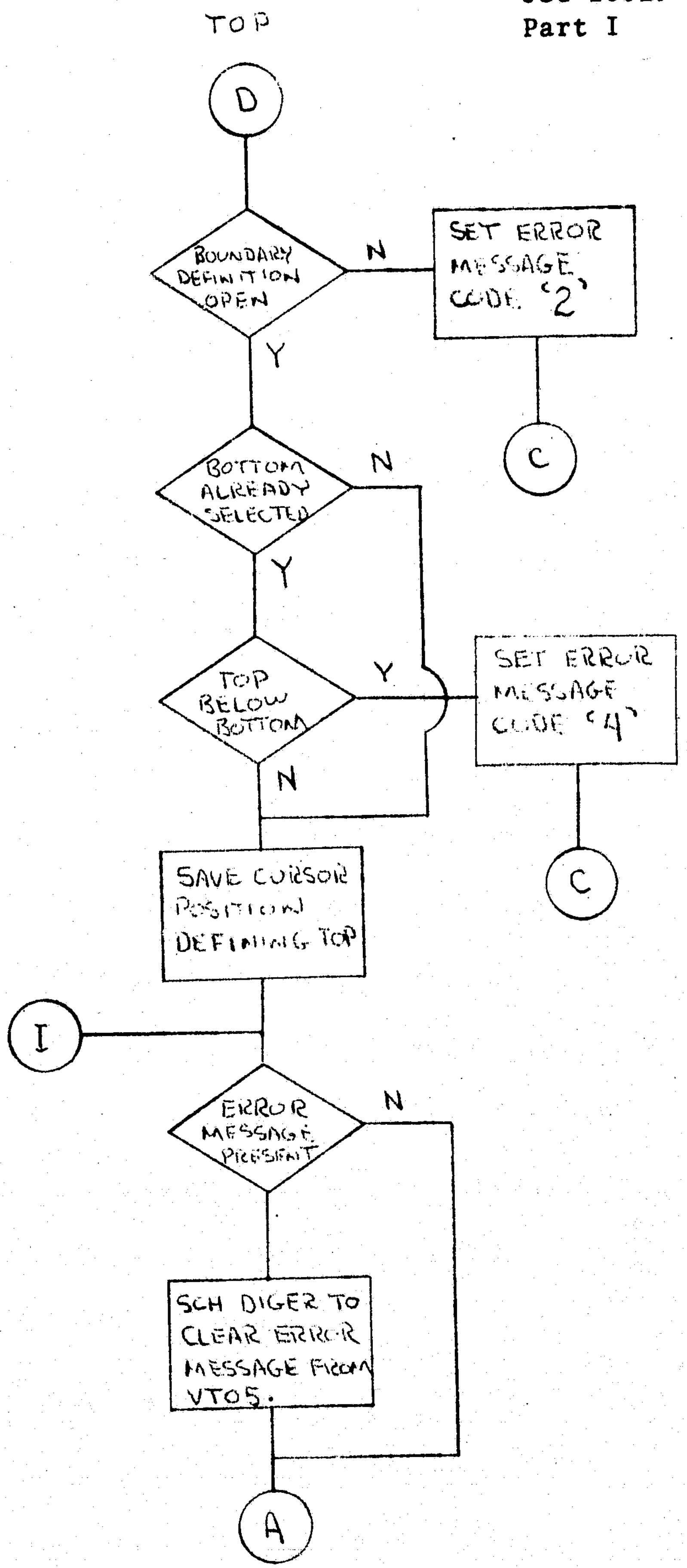


Figure 4-3 (4 of 10)

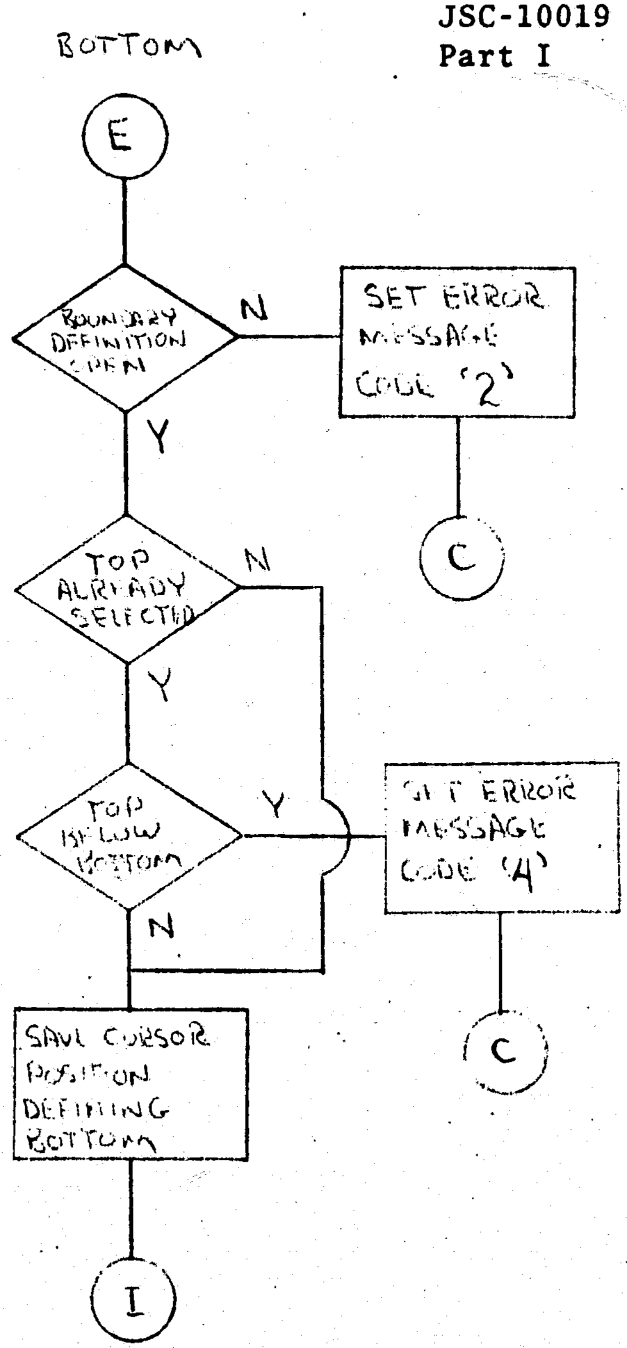


Figure 4-3 (5 of 10)

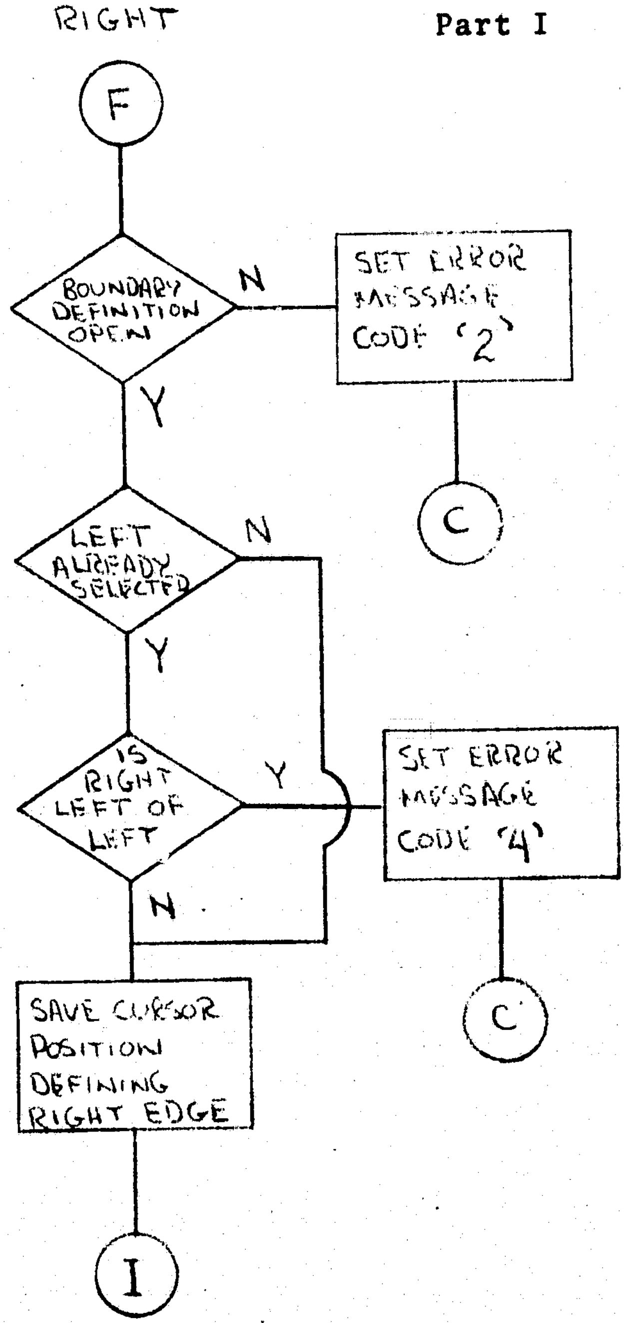


Figure 4-3 (6 of 10)

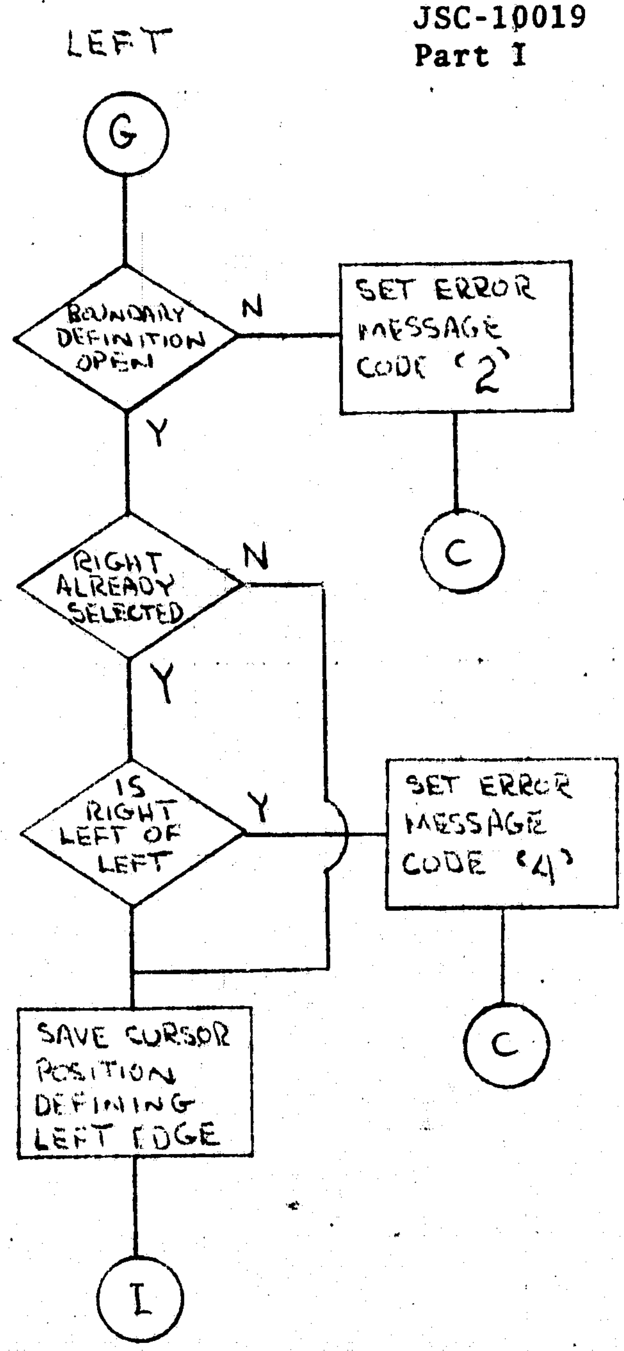


Figure 4-3 (7 of 10)

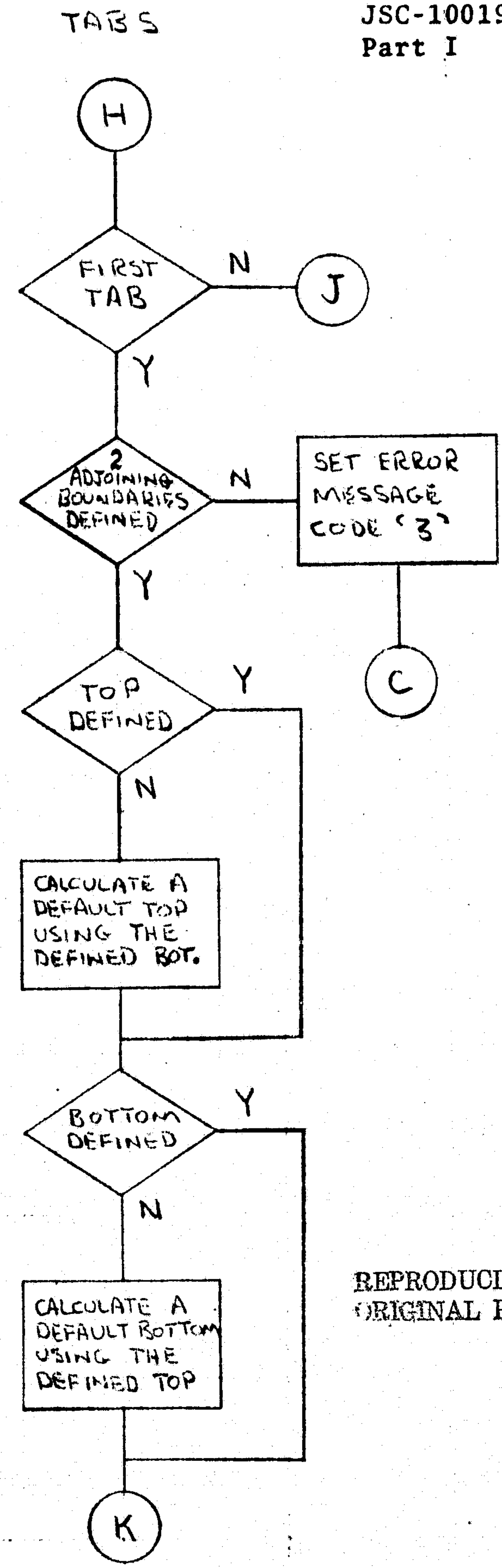


Figure 4-3 (8 of 10)

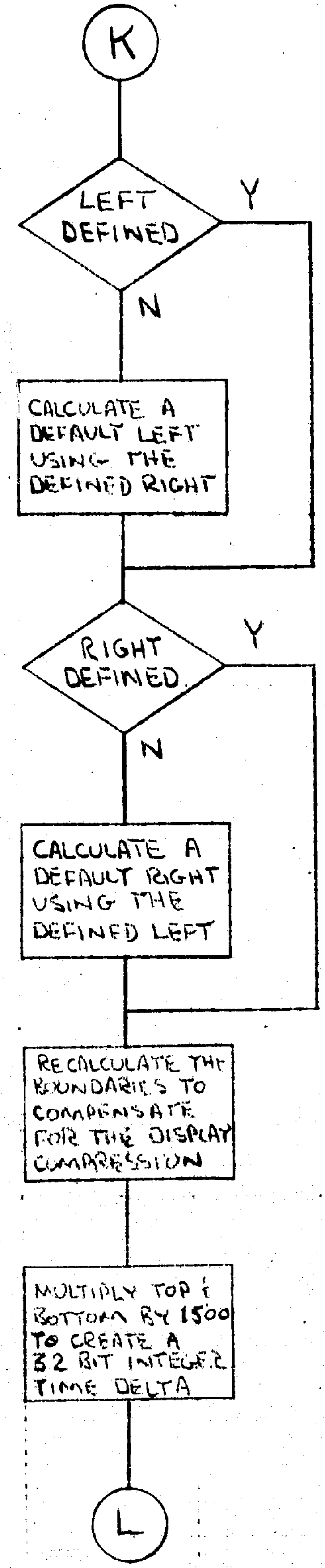


Figure 4-3 (9 of 10)

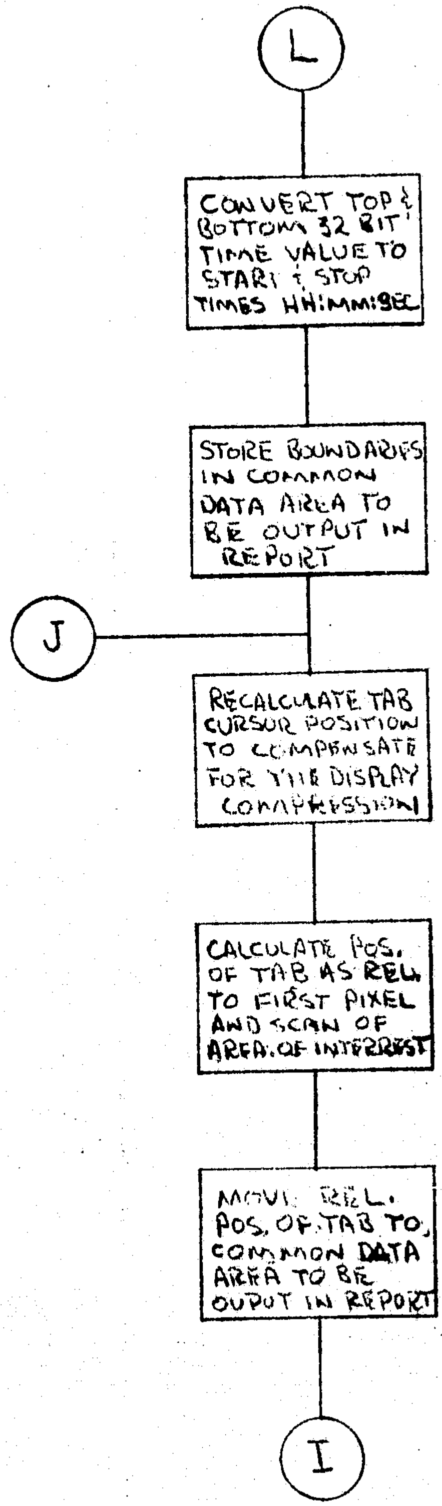


Figure 4-3 (10 of 10)

THIS TABLE IS GENERATED DURING THE SCREENING PASS BY DETERMINING THE CURSER POSITIONS WHEN THE HRCD OPERATOR INITIATES INTERRUPTS. IT IS USED BY THE EDIT RUN TO INITIALIZE THE HARDWARE INTERFACE AND POSITION THE 14-TRACK TAPE AND IS 44_8 BYTES LONG

0	START TIME (DAYS)
+2	START TIME (HOURS)
+4	START TIME (MINUTES)
+6	START TIME (SECONDS)
+10	STOP TIME (DAYS)
+12	STOP TIME (HOURS)
+14	STOP TIME (MINUTES)
+16	STOP TIME (SECONDS)
+20	START SAMPLE NO.
+22	STOP SAMPLE NO.

Figure 4-4 Area of Interest Delimiters (IAREA)

TABLE 4-3

INTER STORAGE REQUIREMENTS

FLAG: IDHFLG (1 WORD)

IDHFLG

2^0 IS SET IF UPPER BOUNDARY DEFINED, CLEAR IF NOT DEFINED

2^1 IS SET IF LOWER BOUNDARY DEFINED, CLEAR IF NOT DEFINED

2^2 IS SET IF RIGHT EDGE DEFINED, CLEAR IF NOT DEFINED

2^3 IS SET IF LEFT EDGE DEFINED, CLEAR IF NOT DEFINED

2^4 IS SET IF BOUNDARY DEFINITION CLOSED, CLEAR IF OPEN

2^5 IS SET IF SUFFICIENT DATA EXISTS FOR TABS, CLEAR IF INSUFFICIENT DATA

2^6 IS SET IF ERROR MESSAGE IS PRESENT, CLEAR IF NOT PRESENT

IDHFLG + 1

0 = 14-TRACK NOT HALTED

377 = 14-TRACK IS HALTED

TABLES

THUMB = 3 WORDS (LEGAL INTERRUPT CODES)
BINDIT = 3 WORDS (INTERRUPT ROUTINE JUMP TABLE)
TABCNT = 1 WORD (NO. OF TABS LOCATED; 16 MAXIMUM)
TOPV = 1 WORD (TOP BOUNDARY SAVE LOCATION)
BOTV = 1 WORD (BOTTOM BOUNDARY SAVE LOACTION)
RHTV = 1 WORD (RIGHT EDGE SAVE LOCATION)
LFTV = 1 WORD (LEFT EDGE SAVE LOCATION)

PROGRAM CODE (INTER)

525-LINE COLOR DISPLAY INTERRUPT HANDLER OF 2206_8 LOCATIONS (BYTES)

- 30 - Cursor on point of interest
- 99 - Terminate screening run.

B. INTER uses interrupt code to index to the proper processing routine and to determine whether the desired function can be completed at this time. The following order of steps must be adhered to by the display operator.

1. Halt the 14-track
2. Define at least two adjoining boundaries. Points of interest cannot be located without them.
3. Define points of interest; boundary redefinition is illegal once any point is located within the window.

C. Error Messages. The screening run can be terminated at any time. The following error messages will be output on the first line of the VT05 when a sequence error is detected. Error messages are cleared with the next legal interrupt received.

1. ILLEGAL INTERRUPT VALUE. An operator interrupt was issued with an undefined thumb-wheel value.
2. INT ISSUED OUT OF SEQUENCE. The three steps defined above were not followed.
3. INSUFFICIENT DATA FOR TABS. An attempt was made to define a point of interest without two adjoining boundaries being defined.
4. ILLEGAL BOUNDARY DEFINITION. An attempt was made to define inversed boundaries (e.g., defining a right boundary to the left of an already defined left boundary).

D. INTER processes legal operator interrupts to define the boundaries of the area of interest to be edited to 9-track tape. The start/stop time pixel window is formatted and output as the last page of the data quality report following termination of the screening run (see paragraph 4.2,G).

INTER communicates with the rest of the SEDS preprocessor via flags and common storage areas.

SECTION 5

COMMON DATA AREAS

All data and constant storage areas required for SEDS processing was added to the D-space modules IDATA and DISDAT.

For IDATA, see the listing, "SEDS DATA POOL." For DISDAT, see listing.